

PHP 开发实战宝典

◎ 潘凯华 邹天思 等编著 ◎

在线服务：**模块库、案例库、题库、素材库、答疑服务**

视频讲解：**43小时**同步语音视频讲解

模块实战：**49个**实战模块，**55项**实验

项目实战：**8个**项目开发实战案例



DVD语音视频教学光盘：

- 43小时同步视频讲解，现场聆听专业讲授
- 附实例及其源程序，让学习、分析、调试程序更方便
- 赠3个项目案例的视频及源程序，方便拓展训练

本书特色：

- 272个小型实例巩固各项必备技术
- 49个实战模块、55项实验提高综合应用能力
- 5个项目案例提高项目开发的熟练度



在线服务方式：

在线服务网站：www.mingribook.com
在线QQ服务：100310267 服务热线：400-675-1066

清华大学出版社

软件开发实战宝典

PHP 开发实战宝典

潘凯华　邹天思　等编著

清华大学出版社
北　京

内 容 简 介

本书从初学者角度出发，以基础知识为“基石”，以核心技术和高级应用为“梁柱”，最终通过实战项目来检验成果，这就是本书的编写思路。本书突出“基础”、“全面”、“深入”，强调“实战”效果。在介绍技术的同时，各章都提供有示例或稍大一些的实例，同时在各章的结尾通过几个小型项目来综合应用本章所讲解的知识，做到理论与实践相结合。最后提供 5 个完整的项目实例，详细讲述从前期规划、系统设计，到项目开发的全部实现过程。

全书共分 29 章，包括初识 PHP，PHP 环境搭建，PHP 语言基础，流程控制语句，函数，字符串，正则表达式，PHP 数组，PHP 与 Web 页面交互，PHP 与 JavaScript 交互，日期和时间，会话管理，图形图像，文件系统，面向对象，PHP 加密技术，MySQL 数据库基本操作，phpMyAdmin 图形化管理工具，使用 MySQL 函数操作数据库，ADODB 类库，程序调试与错误处理，XML 技术，Ajax 技术，Smarty 模板技术，博客管理系统，应用 Smarty 模板开发电子商务平台，办公自动化系统，供求信息网，图书借阅管理系统等内容，最后还在附录中介绍了 PHP 的编码规则。所有知识都结合具体实例进行介绍，对涉及的程序代码给出了详细的注释，可以使读者轻松领会 PHP 程序开发的精髓，快速提高开发技能。

本书适合有志于从事软件开发的初学者、高校计算机相关专业学生和毕业生，也可作为软件开发人员的参考手册，或者高校的教学参考书。

本书提供大量的模块实战和项目实战案例，配备 43 小时视频讲解光盘，提供全部实例源代码，习题都给出了相应的答案。

图书在版编目（CIP）数据

PHP 开发实战宝典/潘凯华，邹天思等编著. —北京：清华大学出版社，2010.1
（软件开发实战宝典）

ISBN 978-7-302-20980-5

I. P…　II. ①潘…　②邹…　III. PHP 语言-程序设计　IV. TP312

中国版本图书馆 CIP 数据核字（2009）第 163121 号

责任编辑：刘利民　郭　伟
版式设计：魏　远
责任校对：柴　燕
责任印制：
出版发行：清华大学出版社　　**地　　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：203×260　**印　　张：**42.25　**字　　数：**1124 千字
（附 DVD 视频光盘 1 张）
版　　次：2010 年 1 月第 1 版　**印　　次：**2010 年 1 月第 1 次印刷
印　　数：1～5000
定　　价：79.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：034018-01

本书编著者名单

主　编： 赛奎春　刘志铭　宋　坤

编　著： 赛奎春　刘志铭　宋　坤　刘彬彬　刘锐宁　王小科　王国辉　邹天思　梁　水　李钟尉　董大勇　杨　丽　刘中华　陈丹丹　潘凯华　李永强　张振坤　梁　冰　房大伟　吕　双　高春艳　安　剑　孙秀梅　刘　欣　刘龄龄　李　慧

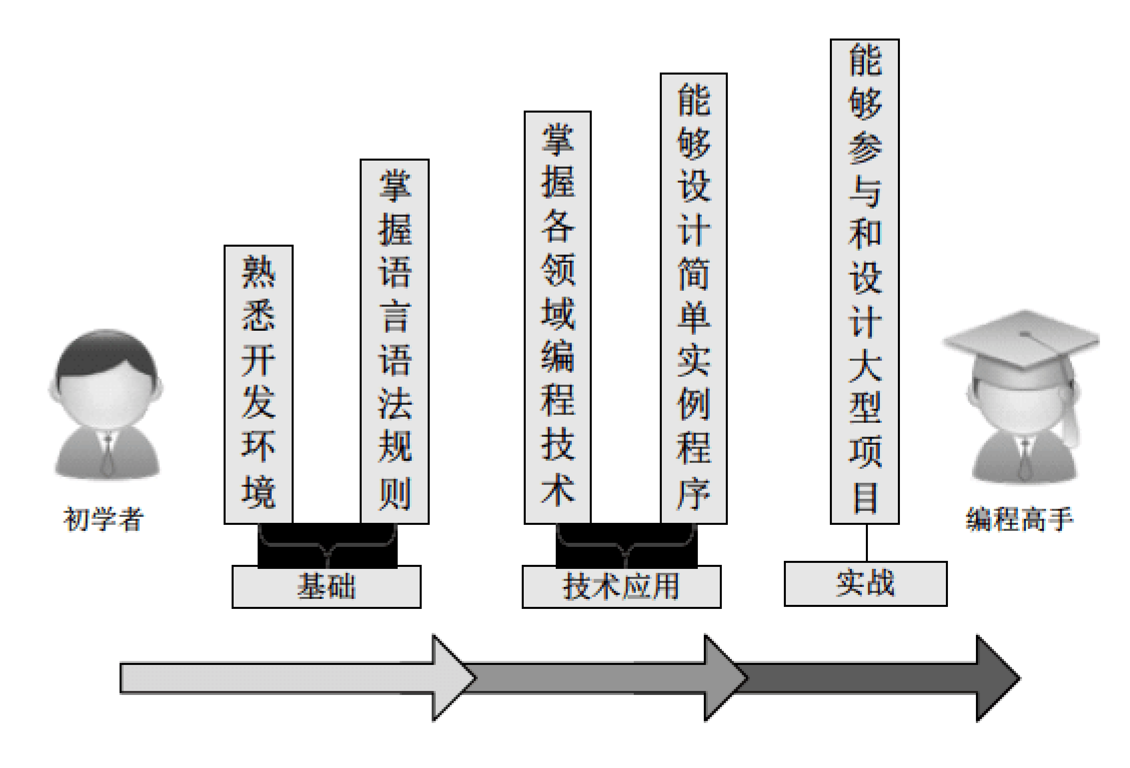
前　言

Preface

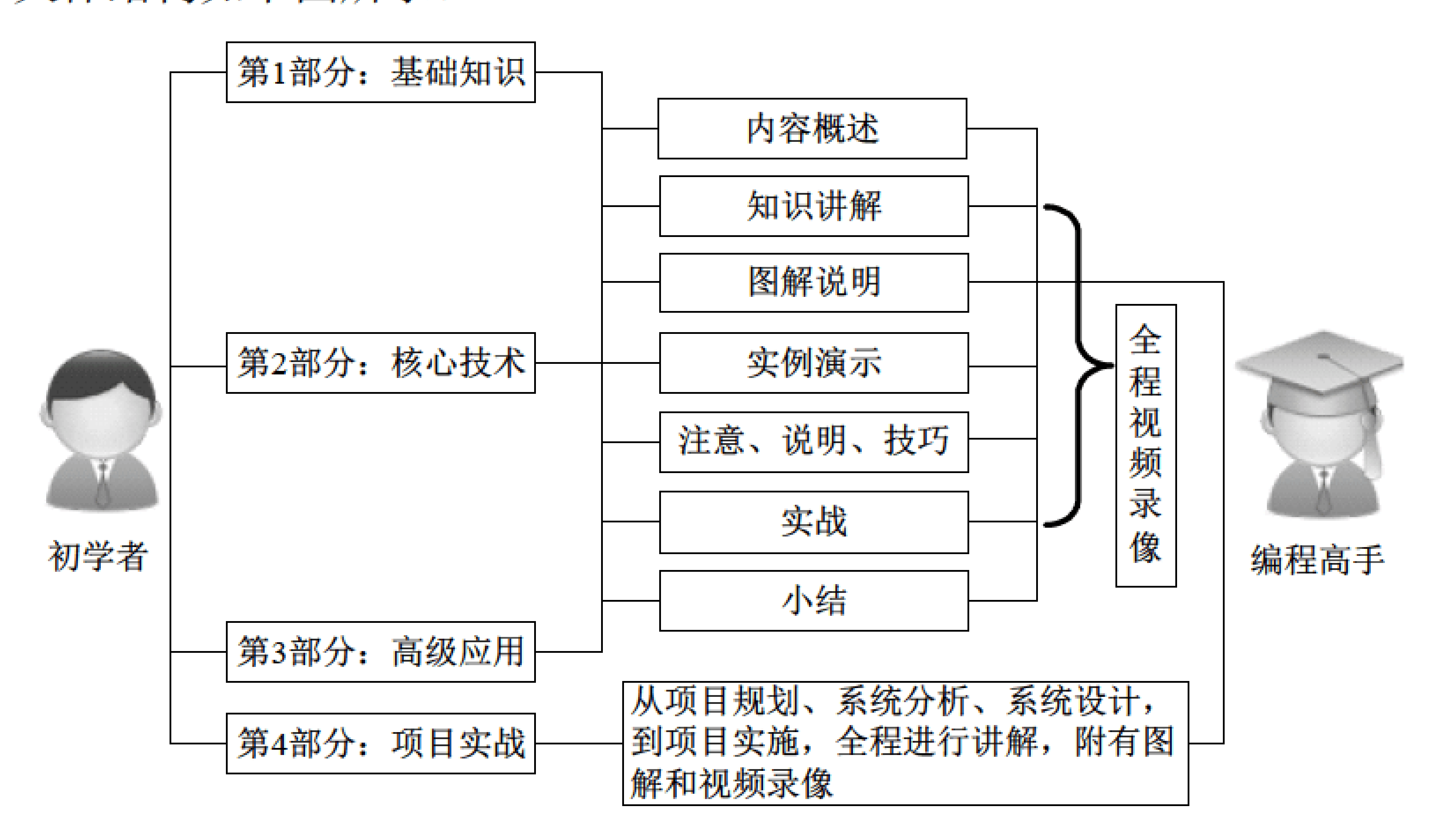
PHP 是全球最普及、应用最广泛的互联网开发语言之一。PHP 语言具有简单、易学、源码开放，可操纵多种主流与非主流的数据库，支持面向对象的编程，支持多种开源框架（如 Zend Framework），支持跨平台的操作，而且完全免费等特点，越来越受到广大程序员的青睐和认同。PHP 目前拥有几百万个用户，其发展速度要快于在它之前的任何一种计算机语言。相信 PHP 一定能够经得起实践的检验，发展成为互联网开发语言中“主流的主流”。

本书内容

从初学程序开发的人员步入到编程高手行列通常需要经历 3 个阶段，如下图所示。



本书中的内容也是按照这一规律精心组织的，将从初学者入门到成为编程高手所必备的各类知识分成 4 个部分，大体结构如下图所示。



第 1 部分：基础知识。通过初识 PHP、PHP 环境搭建、PHP 语言基础、流程控制语句、函数、字符串、正则表达式、PHP 数组、PHP 与 Web 页面交互、PHP 与 JavaScript 交互、日期和时间等章节，并结合大量的图示、示例、综合实例以及录像等，帮助读者快速掌握 PHP 语言，并为以后编程奠定坚实的基础。

第 2 部分：核心技术。介绍了 Cookie 与 Session、图形图像、文件系统、面向对象、PHP 加密技术、MySQL 数据库基本基础、phpMyAdmin 图形化管理工具、使用 MySQL 函数操作数据库、ADODB 类库、PHP 程序调试与异常处理等。学习完这一部分，能够开发数据库应用程序和一些中小型的热点模块。

第 3 部分：高级应用。介绍了 XML 技术、Ajax 技术和 Smarty 模板技术。学习完这一部分，能够开发一些实用的网络程序等。

第 4 部分：项目实战。作者精心挑选并开发了 5 个大型完整的项目。这 5 个项目涵盖了 MySQL 数据库、Smarty 模板技术、ADODB 类库、Ajax 等主流技术。运用软件工程的设计思想，让读者学习如何进行网站项目的实践开发。书中按照编写项目计划书→系统设计→数据库设计→创建项目→实现项目→运行项目→解决开发常见问题→发布网站的过程进行介绍，引领读者一步一步亲身体验开发项目的全过程，使读者可以巩固前面所学的知识和技术，积累项目开发经验。

本书特点

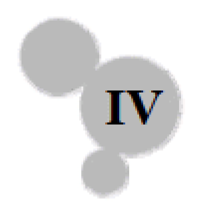
- **配备全程同步语音视频讲解。**
  本书 DVD 光盘提供了覆盖全书的语音视频讲解，读者可以通过视频快速、直观、轻松地学习。
- **提供大量的实战模块、实战案例、实战练习。**
  为了增强读者动手能力，激发学习兴趣，本书提供了大量实战模块、实战案例，读者一定要仔细研读这些模块，并亲自动手调试。
- **提供本书所有实例的源程序。**
  本书光盘提供了所有实例的源程序，读者不必逐字逐句录入，可直接复制，先比猫画虎试着做一遍，可以提高学习效率，模仿是最快的学习方式。
- **本书所有习题和实战都给出了答案，读者可以对照查阅。**

读者对象

- ☑ 有志于软件开发的初学者
- ☑ 准备从事软件开发的求职者
- ☑ 初中级程序开发人员
- ☑ 高等院校计算机相关专业的老师和学生
- ☑ 参与毕业设计的学生
- ☑ 程序测试及维护人员

技术支持与服务

有关于本书的问题，读者可以通过如下方式与我们沟通：



1．登录本技术网站：www.mingribook.com，查阅相关问题或者留言。

2．通过邮箱：tmoonbook@sina.com 或 th_press@263.net。

3．加入 QQ：100310063。

最后，感谢您选择本书，希望本书能成为您编程路上的领航者。

祝读书快乐！

编　者

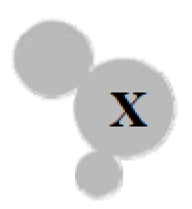


目　录

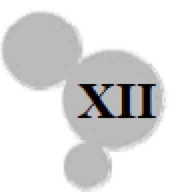
Contents

第 1 部分　基础知识

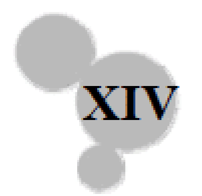




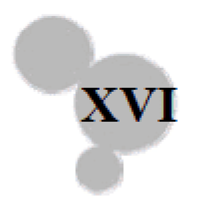
第 2 部分　核心技术



第 3 部分 高级应用



第 4 部分　项目实战





基础知识

第 1 章

初识 PHP

（视频讲解：7 分钟）

PHP 是一种服务器端的嵌入式脚本语言，是一种跨平台、面向对象、HTML 嵌入式的脚本语言。本章将向读者简单介绍 PHP 语言和 PHP 5 的新特性、PHP 的语言优势、常用代码编辑工具，以及学好 PHP 语言的方法等，其主要目的是让读者对 PHP 语言有一个整体的了解和认识，然后再循序渐进地学习，最后达到完全掌握并精通 PHP 语言的目的。

通过阅读本章，可以学习到：

- PHP 是什么
- PHP 5 的新特性
- PHP 的语言优势
- PHP 语言的相关术语
- 学好 PHP 的方法
- PHP 常用的代码编辑工具
- 了解 PHP 的相关学习资源

1.1　PHP 概述

1.1.1　PHP 是什么

PHP 起源于 1995 年，由 Rasmus Lerdorf 开发。PHP 是 Hypertext Preprocessor（超文本预处理器）的缩写，是一种开源、跨平台、独立于架构、解释型、面向对象、快速安全、简单易学、性能优越的 Web 服务器端动态网页开发语言。目前已有超过 2200 万个网站、1.5 万家公司、450 万程序开发人员在使用 PHP 语言，它是目前动态网页开发中使用最为广泛的语言之一。

1.1.2　PHP 5 的新特性

PHP 5 中的对象已经进行了较系统、较全面的调整，下面着重讲述 PHP 5 中新的对象模式。

（1）PHP 5 采用全新的内存管理机制，使其在多线程的环境下可以更有效地运行。

（2）在 PHP 5 全新的面向对象模型中，引入了全新的面向对象机制。

☑　构造函数和析构函数。
☑　对象的引用。
☑　对象的克隆（clone）。
☑　对象中的私有、公共及受保护模式（public/private 和 protected 关键字）。
☑　接口（Interface）。
☑　抽象类。
☑　__call。
☑　__set 和__get。
☑　静态成员。

（3）引入“try…catch”异常处理机制。

（4）全新的 XML 支持。

1.1.3　PHP 的语言优势

PHP 起源于自由软件，即开放源代码软件，使用 PHP 进行 Web 应用程序的开发具有以下语言优势。

（1）安全性高：PHP 是开源软件，每个人都可以看到所有 PHP 的源代码，程序代码与 Apache 编译在一起的方式也可以让它具有灵活的安全设定，PHP 具有了公认的安全性能。

（2）跨平台：PHP 几乎支持所有的操作系统平台（如 Win32 或 UNIX/Linux/Macintosh/FreeBSD/OS2 等），并且支持 Apache、IIS 等多种 Web 服务器，并以此广为流行。

（3）支持广泛的数据库：可操纵多种主流与非主流的数据库，如 MySQL、Access、SQL Server、Oracle、DB2 等，其中 PHP 与 MySQL 是现在最佳的组合，它们的组合可以跨平台运行。

（4）简单易学：PHP 嵌入在 HTML 语言中，以脚本语言为主，内置丰富函数，语法简单、书写



容易，方便学习掌握。

（5）执行速度快：占用系统资源少，代码执行速度快。

（6）开发成本低：在流行的企业应用 LAMP 平台中，Linux、Apache、MySQL 和 PHP 都是免费软件，这种开源免费的框架结构可以为网站经营者节省很大一笔开支。

（7）模板化：实现程序逻辑与用户界面分离。

（8）支持面向对象：支持面向对象和过程的两种风格开发，并可向下兼容。面向对象编程（OOP）是当前的软件开发趋势。PHP 对 OOP 提供了良好的支持，可以使用 OOP 的思想来进行 PHP 的高级编程，这对于提高 PHP 编程能力和规划好 Web 开发架构都非常有意义。

（9）内嵌 Zend 加速引擎，性能稳定快速。

（10）应用范围广：PHP 技术在 Web 开发的各个方面应用得非常广泛。世界上很多大公司都采用了 PHP 技术，如德意志银行的交易系统，华尔街的股票在线买卖，汉莎航空公司的票务处理，甚至美联储、宇航局都采用了 PHP 技术。

1.1.4　PHP 语言的发展

TIOBE 世界编程语言排行榜在一定程度上体现了编程语言在当前的流行趋势。TIOBE 最新公布了 2009 年 3 月的编程语言排行榜，除 Java、C、C++依旧强势占据三甲的位置外，前十名的其他语言排位均有小幅波动。本期值得关注的是 PHP 语言。在本期榜单中，Visual Basic 在坚守了两个月第四名的位置后，本期下降一位，与上期排名第五的 PHP 互换位置。2009 年 3 月份 TIOBE 世界编程语言排行的相关数据说明如图 1.1 所示。

TIOBE 编程语言排行榜衡量了各种编程语言的流行程度。该排行榜每月发布一次，统计数据包括全球范围的软件工程师、培训课程以及第三方供应商，数据来自 Google、MSN 和 Yahoo!等流行搜索引擎。

说明： TIOBE 声称，该排行榜并不是要评选最优秀的编程语言或是统计完成编程量最大的语言，但它可以帮助用户了解自己掌握的编程语言是否跟得上时代的发展，并且在开发新的软件系统需要对编程语言做策略性选择时提供参考。

PHP 是在 1994 年由 Rasmus Lerdorf 创建的，最初只是一个简单的用 Perl 语言编写的统计他自己网站访问者数量的程序。后来重新用 C 语言编写，同时可以访问数据库。1995 年，Personal Home Page Tools（PHP Tools）开始对外发表第一个版本，Rasmus Lerdorf 发布了 PHP 1.0。

此后，越来越多的网站开始使用 PHP，并且强烈要求增加一些特性，如循环语句和数组变量等。在新的成员加入开发行列之后，1995 年的第二版 PHP/FI（Form Interpreter）加入了对 mSQL 的支持，PHP 2.0 的发布，确立了 PHP 在动态网页开发上的地位。

到 1996 年底，有 15000 个网站使用 PHP/FI。

到 1997 年中，使用 PHP/FI 的网站超过 5 万个。

随着 PHP 5.0 的发布和更多对面向对象的支持，PHP 正不断巩固自己在 Web 开发领域的王者地位。

近几年 PHP 的发展呈现上升趋势，如图 1.2 所示，这也说明了 PHP 语言简单、易学、面向对象和安全等特点正在被更多人所认同。相信新的 PHP 语言将会朝着更加企业化的方向迈进，并且将更适合大型系统的开发。

Position Mar 2009	Position Mar 2008	Delta in Position	Programming Language	Ratings Mar 2009	Delta Mar 2008	Status
1	1	=	Java	19.797%	-1.69%	A
2	2	=	C	15.862%	+1.00%	A
3	5	↑↑	C++	10.357%	+1.08%	A
4	4	=	PHP	9.485%	-0.41%	A
5	3	↓↓	(Visual) Basic	8.285%	-3.32%	A
6	7	↑	Python	5.185%	+0.42%	A
7	8	↑	C#	4.321%	-0.19%	A
8	10	↑↑	JavaScript	3.507%	+1.27%	A
9	6	↓↓↓	Perl	3.419%	-2.79%	A
10	9	↓	Delphi	2.710%	-0.09%	A
11	11	=	Ruby	2.601%	+0.74%	A
12	12	=	D	1.809%	+0.62%	A
13	13	=	PL/SQL	1.042%	+0.06%	A
14	22	↑↑↑↑↑↑↑↑	Logo	0.964%	+0.65%	A-
15	14	↓	SAS	0.796%	-0.15%	A

图 1.1　2009 年 3 月 TIOBE 世界编程语言排名

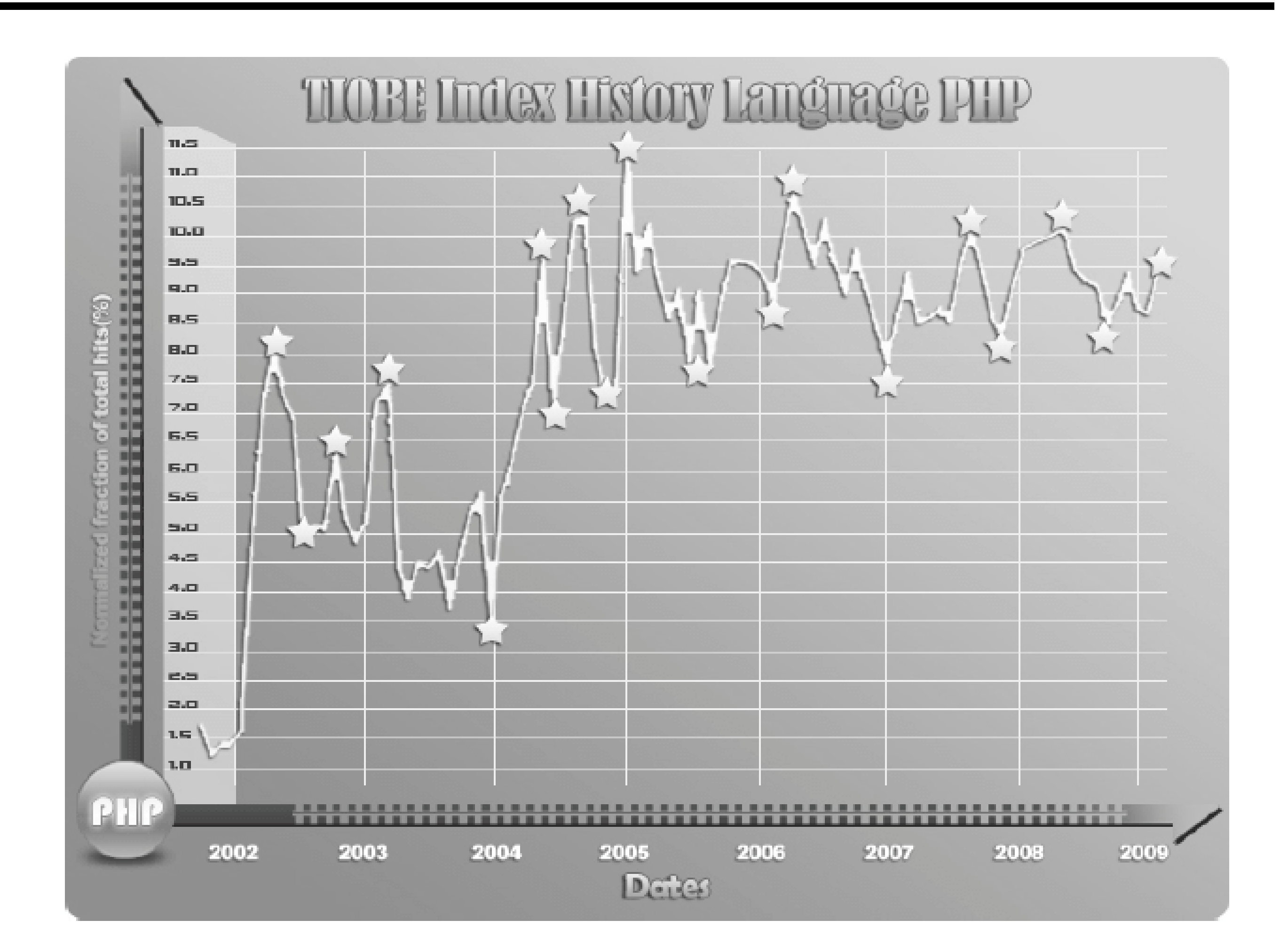


图 1.2　PHP 近几年的语言走势图

说明：图 1.1 和图 1.2 中的数据摘自 http://www.tiobe.com 网站。

PHP 凭借其代码开源、完全免费和安全性高等特性，正吸引着越来越多的 Web 开发人员。主流 PHP 5 的诞生、Zend II 引擎的采用，面向对象的支持以及模板化使得 PHP 编程进入了一个新时代，用户数量成稳步上升趋势。PHP 凭借其强大的功能，必将呈现出良好的发展前景。

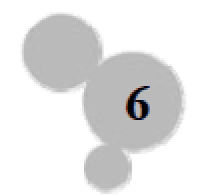
PHP 无处不在，它可以应用于任何领域，并且已拥有几百万个用户，其发展速度要快于在它之前的任何一种计算机语言。PHP 能够给企业和最终用户带来数不尽的好处。据最新数据统计，全世界有超过 2200 万的网站和 1.5 万家公司在使用 PHP 语言，包括百度、雅虎、Google、YouTube、Digg 等著名网站，也包括汉沙航空电子订票系统、德意志银行的网上银行、华尔街在线的金融信息发布系统等，甚至军队系统这类苛刻的环境。除此之外，PHP 也是企业用来构建服务导向型、创造和混合 Web 融于新一代的综合性商业应用的语言，成为开源商业应用发展的方向。

1.2　PHP 与其他语言的比较

就目前的动态网页开发技术而言，除了 PHP 以外，还有 ASP、JSP 和.NET，它们都各有千秋，都有着广泛的用户群，本节将它们进行简单的比较，如表 1.1 所示。

表 1.1　PHP 与其他语言的比较

比 较 项 目	PHP	ASP	JSP	.NET
跨操作系统性	支持	只支持 Win32	支持	只支持 Win32
Web 服务器	多	IIS	很多	IIS
执行效率	快	快	极快	极快
稳定性	高	低	高	高
开发敏捷度	高	高	中	高
支持语言	PHP	VBScript	Java	C#、VB、C++、JScript



续表

比 较 项 目	PHP	ASP	JSP	.NET
函数支持	多	少	中	多
系统安全	高	低	高	高
版本升级	快	慢	慢	一般
难易程度	易	易	难	中

它们都各有所长，可以根据实际需要从中选择一种。不一定要选择最好的，但一定要选择最适合自己的。

1.3　PHP 语言相关术语

PHP 不仅是一门语言，而且已慢慢扩展成为一种行业。专业术语的记忆有助于对新语言的学习，特别是在阅读技术文档时，能帮助理解文档中大量的专业术语，这不仅能加快学习的进度，也能使用户更准确地理解原文的意思，以便在学习过程中少走弯路。下面介绍这门语言涉及的一些常用行业术语。

☑　PHP

PHP 是 Hypertext Preprocessor（超文本预处理器）的缩写，是基于服务器端运行的脚本程序语言。可以实现数据库和网页之间的数据交互。PHP 的独特语法混合了 C 语言、Java 语言和 Perl 语言的特点。PHP 语言可以单独运行，也可以嵌入在 HTML 文件中，这样程序员就可以不必完全依赖 HTML 生成网页。因为 PHP 语言的执行都在服务器端，所以在客户端看不到 PHP 代码。

☑　类

类是对某个对象的定义，是属性和方法的集合，是面向对象编程方式的核心和基础，通过类可以将零散的用于实现某项功能的代码进行有效管理。实际上它本身并不是对象，因为它不存在于内存中。当引用类的代码运行时，类的一个新的实例（即对象）就在内存中创建了。通过类可以在内存中创建多个相同类型的对象。

☑　对象

对象是类进行实例化后的产物，是一个实体。对象就是类存在于内存中的实例。通过对类进行实例化，就能得到对应的对象。

☑　函数

函数是一段相对独立的代码，不同的语言有不同的定义方式。在 PHP 中，函数就是有效的 PHP 代码的集合。在设计复杂程序时，可以把程序分为几个函数，不同的函数完成不同的功能。其优点在于各部分充分独立，不仅可以使程序易读、易维护，还可以对代码进行复用。

☑　Session

Session 用中文来解释就是会话期。在计算机专业术语中，Session 是指一个终端用户与交互系统进行通信的时间间隔，通常指从注册进入系统到注销退出系统所经过的时间。因此，Session 实际上是一个特定的时间概念。Session 是将 Session 的信息保存在服务器上，并通过一个 Session ID 来传递客户端的信息。服务器在接收到 Session ID 后根据这个 ID 来提供相关的 Session 信息资源。由于 Session 为服务器存储，远程用户没办法修改 Session 文件的内容。

☑ Cookie

Cookie 是一种在远程浏览器端储存数据并以此来跟踪和识别用户的机制。简单地说，Cookie 是 Web 服务器暂时存储在用户硬盘上的一个文本文件，并随后被 Web 浏览器读取。在 PHP 中通过 setcookie()函数创建 Cookie。在创建 Cookie 之前，首先必须了解的是，Cookie 是 HTTP 头标的组成部分，而头标必须在页面其他内容之前发送，它必须最先输出，即使在 setcookie()函数前输出一个 HTML 标记或 echo 语句，甚至一个空行都会导致程序出错。

☑ 常量

常量，即不可变的数值或数据项。常量的值通常是在程序运行前定义的，在整个程序运行的过程中，其值是不会改变的。一个常量由英文字母、下划线和数字组成，但数字不能作为首字母出现。常量默认为大小写敏感。按照惯例，常量标识符总是大写的。

☑ 变量

和很多语言不同，在 PHP 中使用变量之前不需要声明变量（PHP 4 之前需要声明变量）。PHP 中的变量用一个美元符号“$”后面跟变量名来表示。PHP 中的变量名是区分大小写的。

☑ 引用

在 PHP 中引用就是用不同的名字访问同一个变量内容。变量名和变量内容是不一样的，因此同样的内容可以有不同的名字。

☑ 构造函数

构造函数又称为构造器，是类中的一个特殊函数。在 PHP 5 中，构造函数使用统一的名称“__construct”进行声明。在使用 new 操作符创建对象时，构造函数将会被自动调用。如果一个类没有构造函数，则调用基类的构造函数，如果有则调用当前类构造函数。一般用于在创建对象时进行初始化操作。

☑ 析构函数

PHP 5 引入了析构函数的概念，通过定义“_destruct”函数来建立析构函数。析构函数与构造函数的功能正好相反，它用于销毁对象。析构函数会在到某个对象的所有引用都被删除或者当对象被显式销毁时执行。

在 PHP 语言的学习过程中，读者应该加强对 PHP 专业术语和其他通用专业术语的学习和积累。

1.4　资深程序员谈如何学习 PHP

怎样学好 PHP 语言，这是所有 PHP 程序员都会考虑的问题。其实，每种语言的学习方法都大同小异，相信大家选择 PHP 语言并不是因为它的背景和悠久历史，更重要的是它的实用性。

下面是笔者结合多年的开发实践所总结出来的学习经验，与所有 PHP 程序开发者分享：

（1）熟悉 HTML/CSS/JavaScript 等网页基本元素，完成阶段可自行制作完整的网页，对元素属性达到熟悉程度。

（2）学会配置 PHP 的开发环境，并选择一种适合自己的开发工具。

（3）理解动态语言的概念，运作机制，熟悉 PHP 语法。

（4）学习如何将 PHP 与 HTML 结合起来完成简单动态页面。



（5）接触 MySQL 数据库，开始设计数据库方面的程序。

（6）不断巩固，熟练使用大部分 PHP 常用函数，并理解 OOP，MySQL 优化，以及模板。

（7）开发一个功能齐全的动态站点。

这套学习方案是一个循序渐进的过程，初学者不要看到上面的短短几行文字就以为学习起来很简单，任何技术其实都不简单，即使是小吃部的美味菜肴也不是一下子就能做成的。

下面解释一下这个学习过程。

任何网站都是由网页组成的，也就是说想完成网站，必须先学会做网页，因此必须要掌握 HTML 语言，才能为今后制作网站打下基础。掌握静态网页的制作技术是学习开发网站的先决条件。

在学习 HTML、CSS 和 JavaScript 的过程中，边学边做是最有效的方式，当然这一方式对于学习 PHP 同样是最有效的。HTML 中的任何元素都要亲自实践，只有熟悉各个元素的效果之后，才会记忆深刻，而对于书本中的知识只是一味地死记硬背是绝对不行的。大部分新手之所以觉得概念难学，大部分是因为“懒惰”，懒惰是阻止一个人走向成功的最大敌人，所以克服懒惰的习惯，才能更快地学好一门语言。

另外，扎实的基础对于一个程序员来说尤为重要，学习知识切记不可急于求成，必须要打好基础，再进阶学习有一定难度的技术，才能轻松地开发网站。而那些不重视“地基”，一心要打造“空中楼阁”的程序员又怎能开发出好的网络程序呢？所以，建议读者多阅读一些基础教材，了解基本的编程知识。

在熟悉了 HTML/CSS/JavaScript 等网页基本元素后，就要开始接触 PHP 了，首先需要做的就是搭建 PHP 环境和选择一款开发工具。

搭建 PHP 环境的方法有很多，可以自定义安装 Apache 服务器、MySQL 服务器和 PHP，然后手动配置环境。但是对于新手来说，最简单的方法就是使用集成化安装包，因为集成化安装包操作起来非常方便，可以直接安装 Apache 服务器、MySQL 服务器、PHP 和 phpMyAdmin，而且不需要用户手动配置，安装完以后即可直接使用。像这种集成化安装包网络上非常多，如使用 AppServ 集成化安装包（第 2 章有详细介绍）。

PHP 的编辑工具有很多，每款编辑工具都有其各自的优势。在编写程序时，一款好的编辑工具会使程序员编写过程更加轻松、有效和快捷，达到事半功倍的效果。在本章 1.5 节中推荐了几款开发工具，读者可以结合自己的需求进行选择。

在理解了 HTML，并熟悉了 PHP 的概念和语法后，进行 PHP 和 HTML 混合编程应该不成问题，在此期间，可以用 PHP 为自己设计一个漂亮的页面，然后在浏览器上输出此时此刻最想说的话，不要觉得幼稚，虽然你输入的是一小段代码，但是对于你的编程之路，可是迈出了整整一大步。

在高兴之余就必须继续努力了，学习 PHP 与数据库的操作。MySQL 数据库可谓是 PHP 的黄金搭档。等到你发奋努力学会了用 PHP 成功地添加、修改、删除数据时，显然距离成功已经指日可待了。巩固了自己的知识，熟悉了 PHP 和 MySQL 开发的要领之后，开发一个留言本，实现一个计数器，你就能充分体会到成就感了。

此时往往是程序员能否精通 PHP 的关键，为什么这么说呢？因为很多程序员当发现自己可以做很多例子，可以动态操作数据库时，就以为学会了 PHP、精通了 PHP。其实，还早呢！你还没有接触 PHP 的精髓 OOP、XML、Ajax 和模板呢！所以，这时候要不懈努力，一一去接触这些技术，精通这些技术之后，你会发现，PHP 的应用是如此的广泛。

总而言之，PHP 的不断升级会让你有学不完的新技术，这里可以用学无止境来形容学习 PHP 的过程。另外，学习 PHP 还有一个重点中的重点——学习编码规范，以此来养成一个良好的编程习惯。

在学习 PHP 的过程中，会接触到大量的函数，PHP 的函数有上千种，下载一个 PHP 中文手册和 MySQL 手册，或者身边备上一两本 PHP 函数类的相关书籍，就可以快速提高编程效率，解决程序中的异常，使其成为你的编程助手。

其实，学习任何一门语言就是一个多实践，多思考，多请教的过程。不要死记语法，特别是对于刚接触 PHP 语言的程序员，掌握好基本语法，反复实践，仅能读懂书本中的内容和技术是不行的，必须动手编写程序代码，并运行程序、分析运行结构，让大脑对学习内容有个整体的认识和肯定，用自己的方式去思考问题，编写代码来提高编程思想。平时多借鉴网上一些好的功能模块，培养自己的编程思想；多向他人请教，学习他人的编程思想；多与他人沟通技术问题，提高自己的技术和见识，这样可以快速地进入学习状态。借鉴别人成功的代码，绝对是有益无害，因为多研究那些经过千锤百炼的经典代码，是提高编程水平的最好方法。另外，目前很多 PHP 书籍都配有视频录像，可以看一些典型的视频录像，从而了解他人的编程思想，快速提高自身的编程水平。

学一门技术最忌急躁，不能遇到技术问题就放弃。遇到问题必须冷静对待，不要让自己的大脑思绪紊乱，保持清醒的头脑才能分析和解决各种问题。可以尝试听歌、散步、玩游戏等活动放松自己。另外还要尝试自己解决，这样可以提高自己的程序调试能力，并能提高解决常见问题的能力。

学习 PHP 不仅要掌握方法，更多的是付出汗水；同时不能轻言放弃，相信自己，相信自己的选择，更要相信自己的能力，如果自己想放弃，不如想想这句话——努力不一定成功，但放弃必定失败！

最后要叮嘱各位的是——抵御诱惑。本章 1.2 节中讲解的 PHP/ASP/JSP/.NET 的对比也许会让你无所适从，你也许学了一半 PHP，又开始打.NET 的主意，或者有人说 Java 很强，这个时候绝对不能动摇，要明确自己的学习目标和方向，选择并锁定一门语言，按照自己的既定目标努力学习和认真研究，直到精通。切记不要哪门都学，哪门都不精。哪怕你真想学，也得学会了 PHP，然后再学。见异思迁是最不可取的，狗熊掰玉米就是这个道理，如果经常中途放弃，只能是一无所获，还浪费了宝贵的时间和经历，得不偿失。

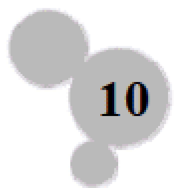
因此，必须要强迫自己完成自己的目标，要有坚持不懈、持之以恒的精神。精通一门语言就是这么坚持下来的。

1.5　常用代码编辑工具

PHP 的开发工具很多，每种开发工具都有其各自的优势。在编写程序时，一款好的编辑工具会使程序员的编写过程更加轻松、有效和快捷，达到事半功倍的效果。因此读者可根据自己的需求有选择性地使用开发工具。

对于一款好的代码编辑工具，除了具备最基本的代码编辑功能外，一个必备的功能就是语法的高亮显示。应用语法的高亮显示，可以对代码中的不同元素采用不同的颜色进行显示，如关键字用蓝色、对象方法用红色标识等。另外，一款好的代码编辑工具应具备格式排版功能，格式排版功能可以使程序代码的组织结构清晰易懂，并且易于程序员进行程序调试，排除程序的错误异常。

下面介绍几款常用的代码编辑工具。



☑ Macromedia Dreamweaver

Macromedia Dreamweaver 是一款专业的网站开发编辑器。它将可视布局工具、应用程序开发功能和代码编辑支持组合在一起，其功能强大，使得各个层次的开发人员和设计人员都能够快速创建出吸引人的、标准的网站和应用程序。它采用了多种先进的技术，能够快速高效地创建极具表现力和动感效果的网页，使网页创作过程简单无比。同时，Macromedia Dreamweaver 提供了代码自动完成功能，不但可以提高编写速度，而且减少了错误代码出现的几率。Macromedia Dreamweaver 既适用于初学者制作简单的网页，又适用于网站设计师、网站程序员开发各类大型应用程序，极大地方便了程序员对网站的开发与维护。

Macromedia Dreamweaver 从 MX 版本开始支持 PHP+MySQL 的可视化开发，对于初学者确实是比较好的选择，因为如果是一般性开发，几乎是可以不用一行代码也可以写出一个程序，而且都是所见即所得的。所包含的特征包括：语法加亮、函数补全，形参提示、全局查找替换、处理 Flash 和图像编辑等。同时，可以为 PHP、ASP 等脚本语言提供辅助支持。

另外，Macromedia Dreamweaver 从第 8 版开始，提供代码折叠功能，可以将一个代码块，如一个方法或者一个类的代码块折叠起来用两行代替，需要时可以再展开。

下载地址：http://www.adobe.com/downloads/。

注意： Macromedia Dreamweaver MX 是首选的网页开发工具。本书所介绍的网页和实例都是使用 Macromedia Dreamweaver 8 编辑的。

☑ Eclipse

Eclipse 是一款支持各种应用程序开发工具的编辑器，为程序设计师提供了许多强悍的功能。它支持多语言的关键字和语法加亮显示，支持查询结果匹配部分在编辑器中的加亮显示，支持代码格式化功能，还具备强大的调试功能，可以设置断点，使用单步执行方法执行源代码。

官方网站：http://www.eclipse.org。

☑ ZendStudio

ZendStudio 是目前公认的最强大的 PHP 开发工具，具备功能强大的专业编辑工具和调试工具，包括编辑、调试、配置 PHP 程序所需要的客户及服务器组件，支持 PHP 语法加亮显示，尤其是功能齐全的调试功能，让 PHP 错误不再可怕。ZendStudio 是一款收费软件，不过可以免费下载试用版。

下载地址：http://www.zend.com/store/products/zend-studio.php。

☑ PHPEdit

PHPEdit 是一款 Windows 操作系统下优秀的 PHP 脚本 IDE（集成开发环境）。该软件为快速、便捷地开发 PHP 脚本提供了多种工具，其功能包括：语法关键词高亮；代码提示、浏览；集成 PHP 调试工具；帮助生成器；自定义快捷方式；150 多个脚本命令；键盘模板；报告生成器；快速标记；插件等。

官方网站：http://phpedit.svoi.net。

☑ UltraEdit

UltraEdit 是能够满足用户一切编辑需要的编辑器。UltraEdit 是一套功能强大的文本编辑器，可以编辑文本、十六进制、ASCII 码，可以取代记事本程序，内建英文单字检查、C++及 VB 指令突显，可同时编辑多个文件，而且即使开启很大的文件速度也不会变慢。软件附有 HTML 标签颜色显示、搜寻替换以及无限制的还原功能，用户喜欢用其来修改 EXE 或 DLL 文件。

官方网站：http://www.ultraedit.com.cn。

1.6 学习资源

视频讲解：光盘\TM\Video\1\PHP 的学习资源.exe

作为一种被广泛使用的 Web 应用开发语言，PHP 有着极其丰富的学习资源。

1.6.1 PHP 参考手册

学习 PHP 语言，在电脑中必须配备一个 PHP 参考手册，就像在学习汉字时，身边必须配备一本《新华字典》一样。因为 PHP 参考手册是 PHP 语言的“新华字典”，它对 PHP 的函数进行了详细的讲解和说明，并且还给出了一些简单的示例，同时还对 PHP 的安装与配置、语言参考、安全和特点等一些信息进行了介绍。

在 PHP 的 http://www.php.net/docs.php 网站上，提供了 PHP 的各种语言、格式和版本的 PHP 参考手册，用户可以进行在线阅读，也可以打包下载。

PHP 参考手册不但对 PHP 的函数进行了解释和说明，而且还提供了快速查找的方法，让用户可以更加方便地查找到指定的函数。PHP 参考手册（下载版）的运行效果如图 1.3 所示。



图 1.3　PHP 参考手册

1.6.2 PHP 学习讨论

☑ PHP 的官方网站（http://www.php.net）

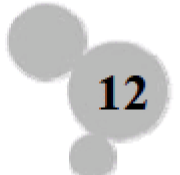
提供 PHP 各版本的下载，以及 PHP 手册下载。

☑ 喜悦国际村（http://www.phpx.com）

目前国内最火爆的 PHP 讨论社区，注册用户数万，PHP 版面讨论激烈，PHP 高级工程师众多。

☑ PHP 5 研究室（http://www.phpv.net）

关注新技术的研究和跟踪，能够了解新技术，同时能够查询到最新版本的手册。



☑　PHPChina（http://www.phpchina.com）

PHPChina 开源社区，有非常多的 PHP 源码，能够从中获取最新的 PHP 信息资源。

☑　Linux 和 UNIX 中文网站（http://www.chinaunix.net）

全球最大的 Linux 和 UNIX 中文网站，其中的 PHP 版块资料极其丰富。

☑　PHPWind（http://www.phpwind.com）

PHPWind 论坛，是国内发展最迅速的论坛，代码质量比较高，执行速度快。

☑　Discuz!（http://www.discuz.com/）

国内最著名的 PHP 论坛，速度快、稳定性好、安全。易于定制，能够支持非常多的在线人数。目前该论坛已彻底免费。

1.7　成 功 案 例

目前，互联网上很多网站的开发都是通过 PHP 语言来完成的，如搜狐、网易和百度等，在这些知名网站的创作开发中都应用到了 PHP 语言。

吉林省明日科技有限公司软件网、图书网、网络学院、编程词典网等也都是用 PHP 语言来完成的。感兴趣的读者可以登录下面的网站，不但可以浏览完全由 PHP 开发的网站，体会 PHP 语言的强大功能，同时也可以通过网站与我们进行交流、互相学习，也可以更深入地了解明日科技，了解我们的编程词典，了解我们的网络学院。

吉林省明日科技有限公司的图书网网址是 http://www.mingribook.com，如图 1.4 所示。

吉林省明日科技的网络学院的网址是 http://www.cccxy.com，如图 1.5 所示。



图 1.4　明日科技图书网首页



图 1.5　明日网络学院网站首页

吉林省明日科技的论坛网网址是 http://www.mingribook.com/bbs/bbs_index.php，如图 1.6 所示。

吉林省明日科技的编程词典网的网址是 http://www.mrbccd.com/index.shtml，如图 1.7 所示。



图 1.6　明日科技论坛网首页



图 1.7　明日科技编程词典服务网首页

1.8　本 章 小 结

本章重点讲述了什么是 PHP，PHP 的语言优势以及应用领域，详细介绍了 PHP 5 的新特性，通过这些内容使读者对 PHP 有一个全面的认识。其次，指导读者学习 PHP 的方法。另外，本章的最后还介绍了常用代码编辑工具、网络上的学习资源以及一些经典的 PHP 成功案例，以方便读者更好地学习 PHP。

第 2 章

PHP 环境搭建

（视频讲解：80 分钟）

PHP 是全球最普及、应用最广泛的互联网开发语言之一。学习任何一门编程语言，在开始学习之前都要首先学会搭建和熟悉开发环境，本章将介绍两种操作系统（Windows 和 Linux）下的 Apache 服务器、MySQL 服务器及 PHP 的安装和配置方法。另外还特别为初学者介绍一种简化安装和配置 PHP 环境的组合包。最后，使用 Macromedia Dreamweaver 8 开发 PHP 第一个实例。

通过阅读本章，可以学习到：

- **如何在 Windows/Linux 下安装和配置 Apache 服务器**
- **如何在 Windows/Linux 下安装和配置 MySQL 服务器**
- **如何在 Windows/Linux 下安装 PHP**
- **如何使用组合包快速搭建 PHP 环境**
- **如何应用开发工具编写、发布和运行第一个 PHP 程序**
- **如何独立解决常见的 PHP 环境安装问题**

2.1 搭建 PHP 开发环境的准备工作

2.1.1 在 Windows 下搭建 PHP 开发环境的准备工作

在 Windows 下搭建 PHP 与安装其他的一些软件工具不同。因为 PHP 是从 Linux 移植过来的一种语言，不仅在开发环境上尽量保留着 Linux 的特点（Apache 是 Linux 下的 Web 服务器，地位就像 Windows 下的 IIS；MySQL 也是 Linux 系统中捆绑的数据库），在安装上也被烙上了 Linux 印记。除了正常的安装操作外，还需要在各自的配置文件（.ini、.conf）中进行专门的设置。

安装之前要准备的安装包有：

☑ Apache_2.2.8-win32-x86-no_ssl.msi。下载地址为 http://httpd.Apache.org/download.cgi。

☑ php-5.2.5-Win32.zip。下载地址为 http://www.php.net/downloads.php。

☑ mysql-essential-5.0.51a-win32.msi。下载地址为 http://www.mysql.com/download/（下载 MySQL 需要注册一个账号）。

2.1.2 在 Linux 下搭建 PHP 开发环境的准备工作

在 Linux 下搭建 PHP 环境比 Windows 下要复杂得多。除了 Apache、PHP 等软件外，还要安装一些相关工具，设置必要的参数。而且，要使用 PHP 扩展库，还要进行编译。如本书中使用到的 SOAP、MHASH 等扩展库。这里给出在 Linux 下安装的必要步骤。如果用户在安装过程中遇到特殊的问题，还需要翻阅 Linux 相关的书籍、手册。

安装之前要准备的安装包有：

☑ httpd-2.2.8.tar.gz。下载地址为 http://www.apache.org。

☑ php-5.2.5.tar.gz。下载地址为 http://www.php.net/downloads.php。

☑ mysql-5.0.51a-Linux-i686.tar.gz。下载地址为 http://www.mysql.com。

☑ libxml2-2.6.26.tar.gz。可在网络上直接搜索该版本进行下载。

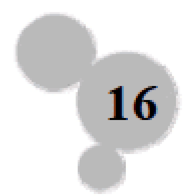
2.2 Apache 服务器的安装和配置

2.2.1 在 Windows 下安装 Apache 服务器

视频讲解：光盘\TM\Video\2\在 Windows 下安装 Apache 服务器.exe

Apache 服务器是全球范围内使用范围最广的 Web 服务软件，超过 50%的网站都在使用 Apache 服务器，它以其高效、稳定、安全、免费（最重要的一点）的优势成为了最受欢迎的服务器软件。

本节主要介绍如何在 Windows 操作系统中安装和配置 Apache 服务器。安装 Apache 服务器前，应到官方网站 http://www.apache.org 下载 Apache 的安装程序 Apache_2.2.8-win32-x86-no_ssl.msi。



在 Windows 下安装和配置 Apache 服务器的操作步骤如下。

（1）双击 Apache_2.2.8-win32-x86-no_ssl.msi 文件，弹出欢迎页面。单击 Next 按钮，进入到许可协议页面。

（2）在许可协议页面，用户需要同意页面中的条款才能继续安装。选中 I accept the terms in the license agreement 单选按钮，页面如图 2.1 所示。单击 Next 按钮进入到下一页面。

（3）本页面是对该程序的一个描述和说明。在了解了相关的信息后，单击 Next 按钮进入到 Server Information 页面。

（4）Server Information 页面需要用户填写域名、服务器名称和管理员 Email。Server Information 页面的填写效果如图 2.2 所示。该页面还有两个单选按钮，如果选中默认的第一个单选按钮，说明该服务器对所有人开放，并且服务器的端口号为 80，这个是推荐选项。第二个单选按钮是指该服务器仅对当前用户开放，并且服务器端口为 8080。这里选中第一个单选按钮。然后单击 Next 按钮进入下一个页面。

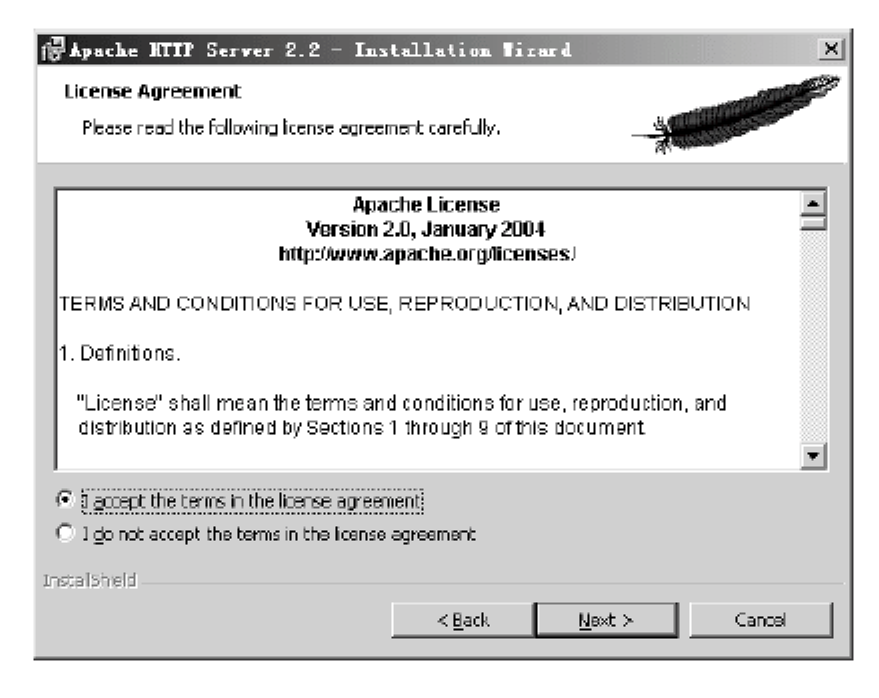


图 2.1　许可协议页面

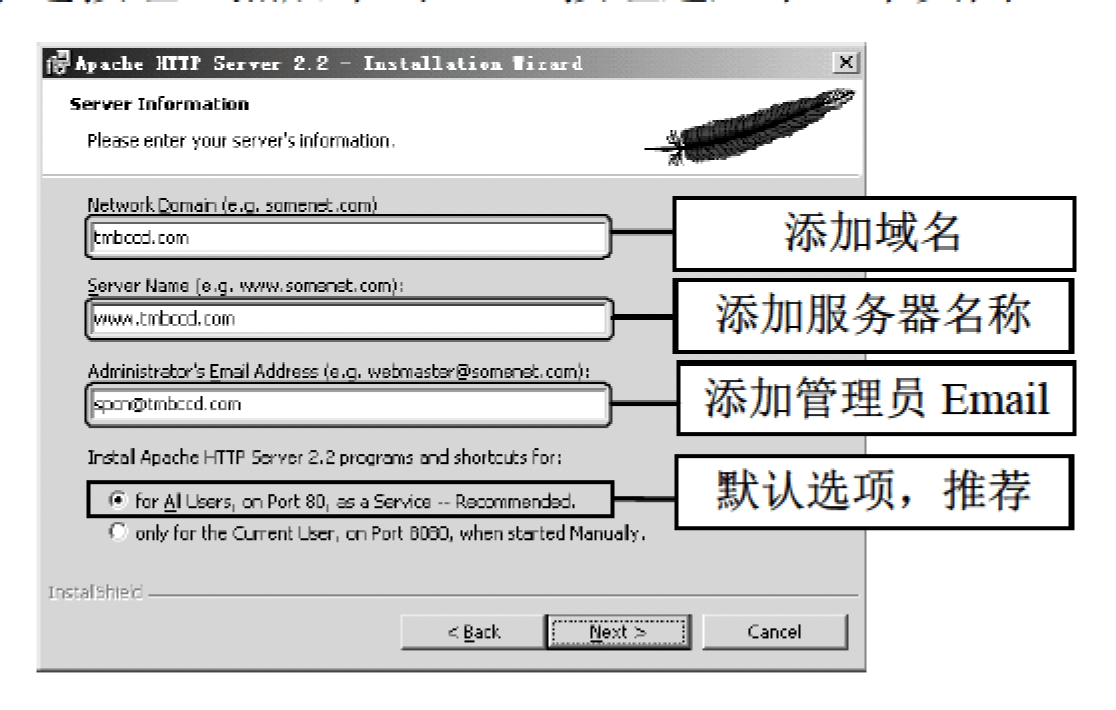


图 2.2　Server Information 页面

注意： 如果用户的机器安装有“Internet 信息服务（IIS）管理器”，那么必须将此项服务停止，因为 IIS 服务器的默认端口号为 80，同 Apache 服务器默认端口号相同。如果 IIS 服务不停止，就会和 Apache 服务器的端口号产生冲突，Apache 服务器将不能成功安装。

（5）如图 2.3 所示页面用于选择安装类型。安装类型分为典型安装和自定义安装，通常保持默认选项即可。单击 Next 按钮，进入到路径选取页面。

（6）在路径选取页面中，单击 Change 按钮可以选择安装路径。这里路径设为“D:\Apache2.2\”，如图 2.4 所示。

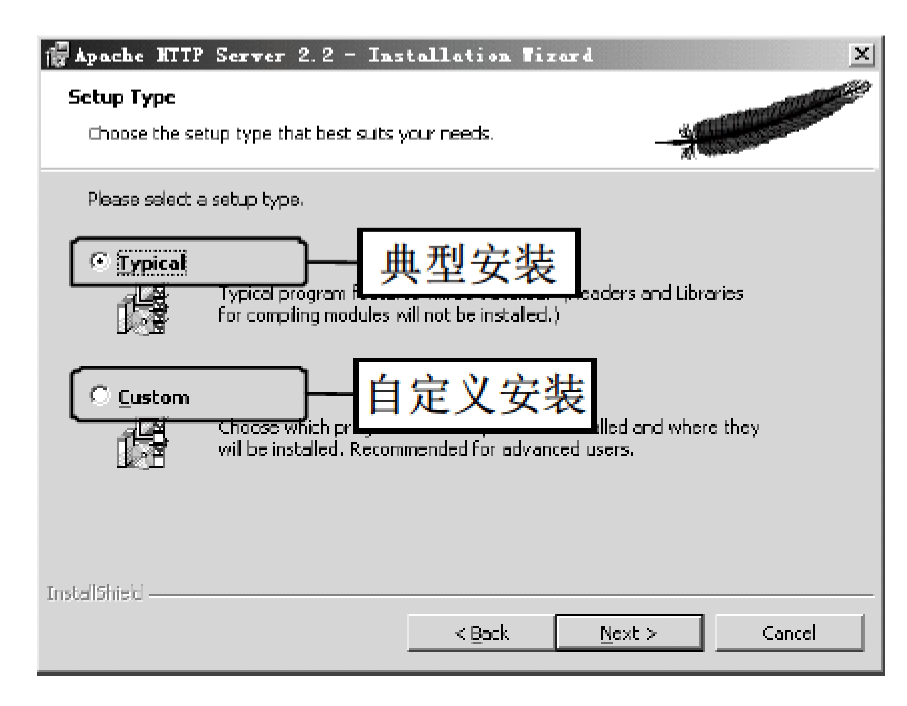


图 2.3　选择安装类型

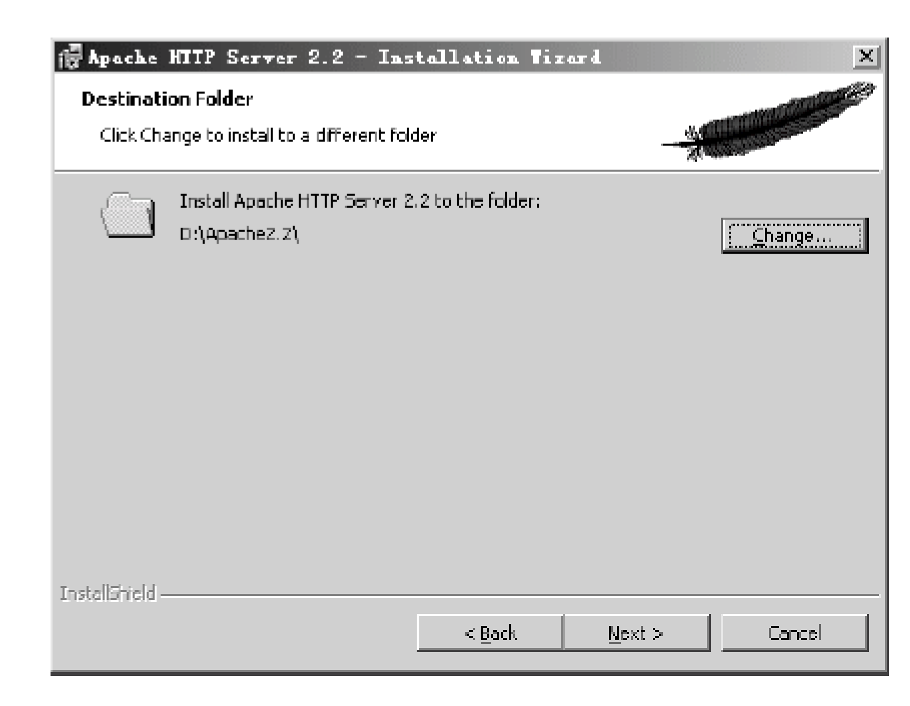


图 2.4　指定路径



（7）单击 Next 按钮进入文件安装页面。这是 Apache 安装的最后一步，程序开始安装文件。安装结束后，单击 Finish 按钮结束安装程序。

（8）安装完成后，Apache 服务器会自动开启。在系统托盘区域将出现一个图标，当前 Apache 服务启动时，图标样式为；服务器未启动时，图标样式为。

单击 Apache 服务器的启动小图标，将会看到服务器的开启与关闭功能；也可以用鼠标右键单击小图标，在弹出的快捷菜单中选择 Open Apache Monitor 命令，打开 Apache 监控程序，其操作效果如图 2.5 所示。

（9）服务器开启后，最后需要测试一下服务器。打开 IE 浏览器页面，在地址栏中输入“http://127.0.0.1/”或“http://localhost/”，按 Enter 键后系统会显示如图 2.6 所示的页面，此时说明 Apache 服务器正式安装成功。

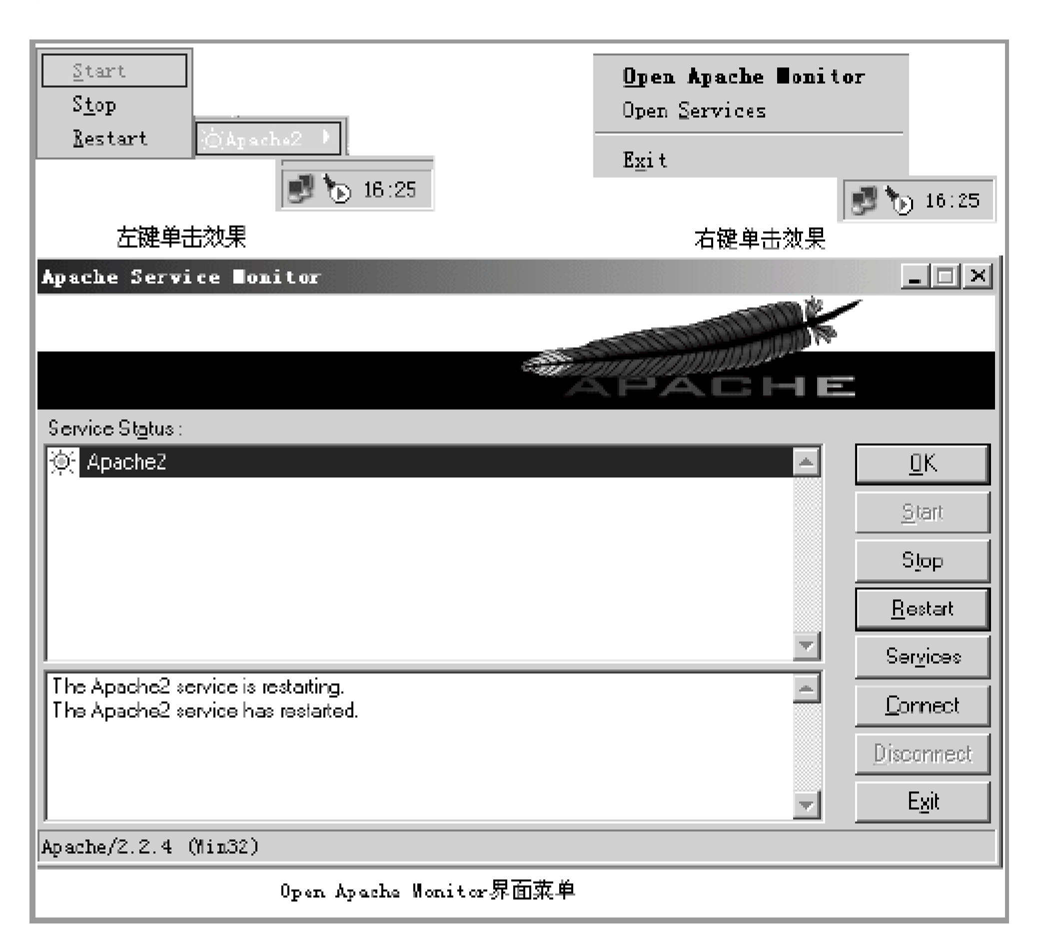


图 2.5　Apache 控制菜单

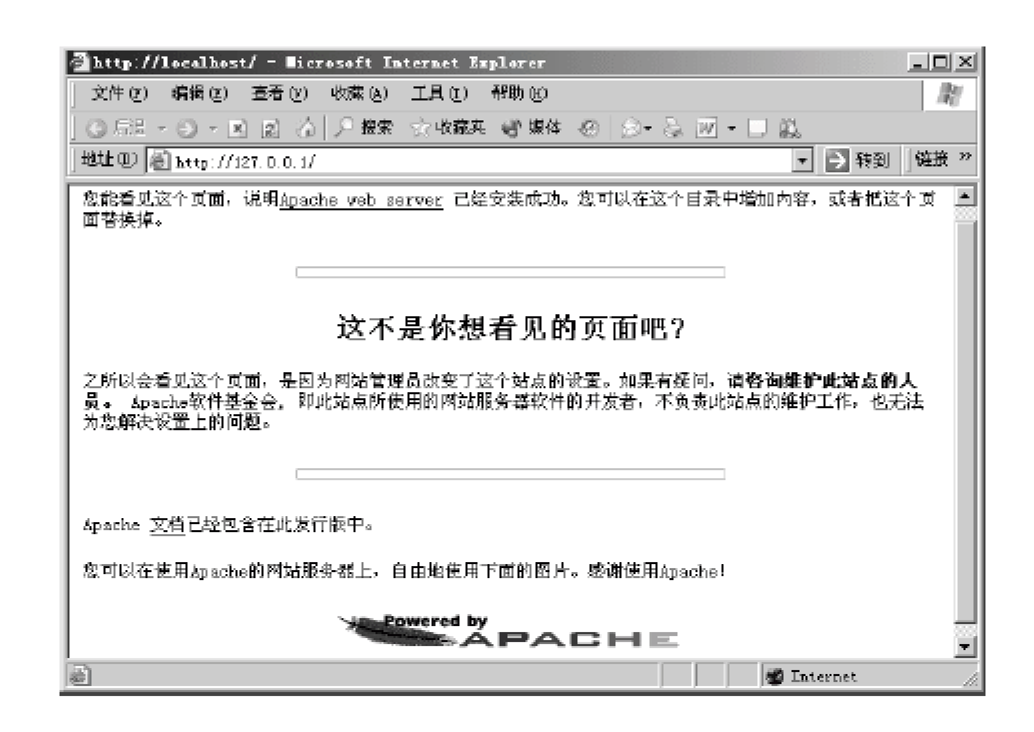


图 2.6　Apache 服务器运行页面

（10）Apache 服务器安装成功后，接下来需要对 Apache 服务器进行配置，以便 Apache 服务器能够识别 PHP 文件。配置 Apache 服务器主要是在 Apache 安装目录下的 conf 子目录中的 httpd.conf 文件中进行的，找到该文件并用记事本等文本编辑器打开该文件。

（11）定位到 LoadModule 配置块，在 LoadModule 的最后添加如下信息：

```
LoadModule php5_module d:\php5\php5Apache2_2.dll
```

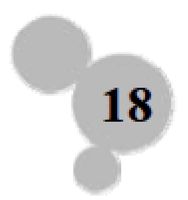
添加后的文件结果如图 2.7 所示。

（12）修改 DocumentRoot 参数可以修改 Apache 服务器主文档的根目录。原根目录的位置是 Apache2.2\htdocs，用户可以任意指定位置。如：

```
DocumentRoot "D:/Webpage"
```

在 DocumentRoot 的下面间隔约 28 行的位置，有一行为<Directory "D:/Apache2.2/htdocs">，修改为<Directory "D:/Webpage">。

注意：DocumentRoot 和这里的参数值要保持一致。



（13）添加 Apache 服务器能够识别的 PHP 扩展名。PHP 的扩展名有.php3、.php4、.php、.phtml 等。这里只推荐使用标准的扩展名.php。添加的代码如下：

```
AddType application/x-httpd-php .php
```

添加位置如图 2.8 所示。

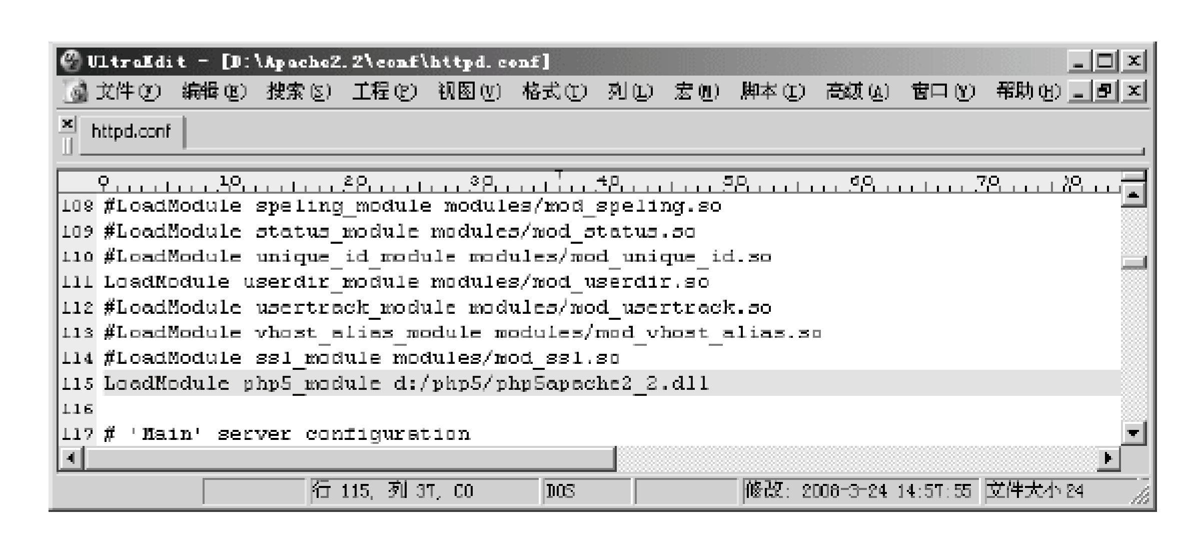


图 2.7　加载 dll 文件

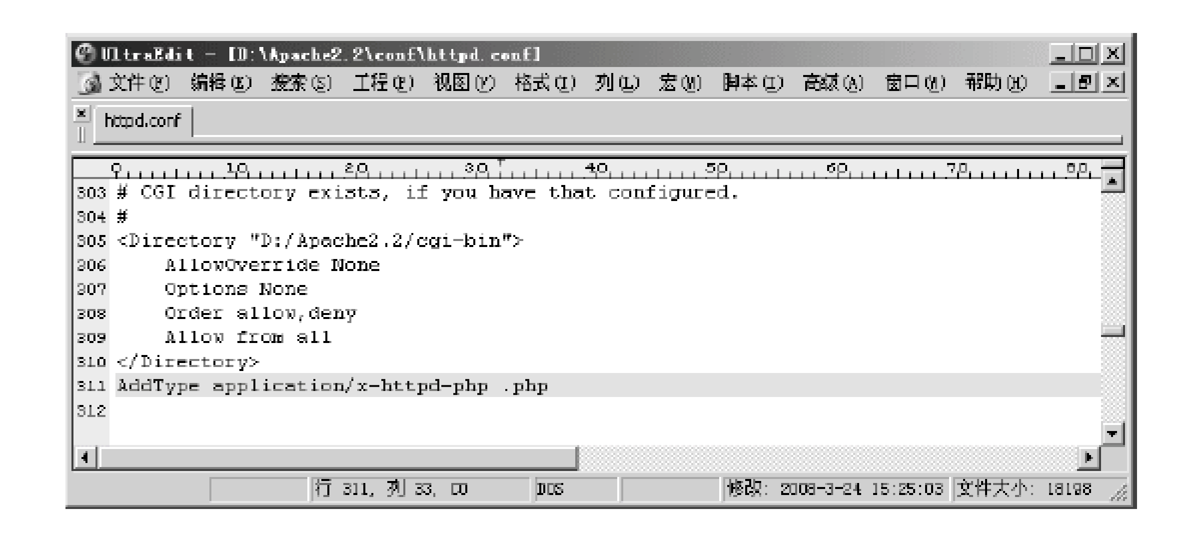


图 2.8　添加 PHP 扩展识别

（14）默认显示页。Apache 的默认显示页为 index.html。也就是说，在服务器未指名文件时，首先查找 index.html，如果找到 index.html，那么服务器就将加载该文件，否则显示目录内的文件列表。在这里添加一个 PHP 默认页：index.php。更改后的代码如下：

```
DirectoryIndex index.html index.php
```

（15）修改 Apache 端口号。Apache 的端口号为 80。修改 Listen 选项的值，即可修改端口号。如改为 82，则更改后的代码如下：

```
Listen 82
```

以上配置完成后，重启 Apache 服务器即可。

注意： 如果用户的计算机上还有 IIS 服务器，那么可能会因为端口冲突而导致 Apache 无法正常开启。解决的办法是改变其中的一个端口号，或者停止 IIS 服务器。

2.2.2　在 Linux 下安装 Apache 服务器

视频讲解：光盘\TM\Video\2\在 Linux 下安装 Apache 服务器.exe

在 Linux 下安装 Apache 需要到官方网站 http://www.apache.org 下载 Linux 下的 httpd2.2.8.tar.gz 的压缩包。

首先需要打开 Linux 终端（Linux 下几乎所有的软件都需要在终端下安装）。在 RedHat 9 的界面中选择“主菜单”/“系统工具”命令，在弹出的菜单中选择“终端”命令。

在 Linux 下安装和配置 Apache 服务器的操作步骤如下：

（1）进入到 Apache 安装文件的目录下，如/usr/local/work。

```
cd /usr/local/work/
```

（2）解压安装包。完成后进入到 httpd2.2.8 目录中。

```
tar xfz httpd2.2.8.tar.gz
cd httd2.2.8
```

（3）建立 makefile，将 Apache 服务器安装到 user/local/Apache2 下。

```
./configure –prefix=/usr/local/Apache2 –enable-module=so
```

（4）编译文件。

```
make
```

（5）开始安装。

```
make install
```

（6）安装完成后，将 Apache 服务添加到系统启动项中，重启服务器。

```
/usr/local/Apache2/bin/Apachectl start >> /etc/rc.d/rc.local
/usr/local/Apache2/bin/Apachectl restart
```

（7）打开 Mozilla 浏览器，在地址栏中输入“http://localhost/”，按 Enter 键后如果看到如图 2.9 所示的页面，说明 Apache 服务器已安装成功。



图 2.9　Linux 下的 Apache 服务器安装页面

2.3　PHP 的安装和配置

2.3.1　在 Windows 下安装 PHP

视频讲解：光盘\TM\Video\2\在 Windows 下安装 PHP.exe

架设基于 PHP 的 Web 服务器，必须安装 PHP。由于 PHP 的代码公开，所以其升级速度较快。安装 PHP 之前应从其官方网站 http://www.php.net/下载最新版本的 PHP 安装程序 php-5.2.5-Win32.zip。

Apache 服务器顺利启动后，接下来安装 PHP 5。在 Windows 下安装和配置 PHP 的操作步骤如下：

（1）将 PHP 5 的安装文件 php-5.2.5-Win32.zip 解压到相应目录，如 c:\php、d:\php5 等。这里将其放到 d:\php5 目录下。目录结构如图 2.10 所示。



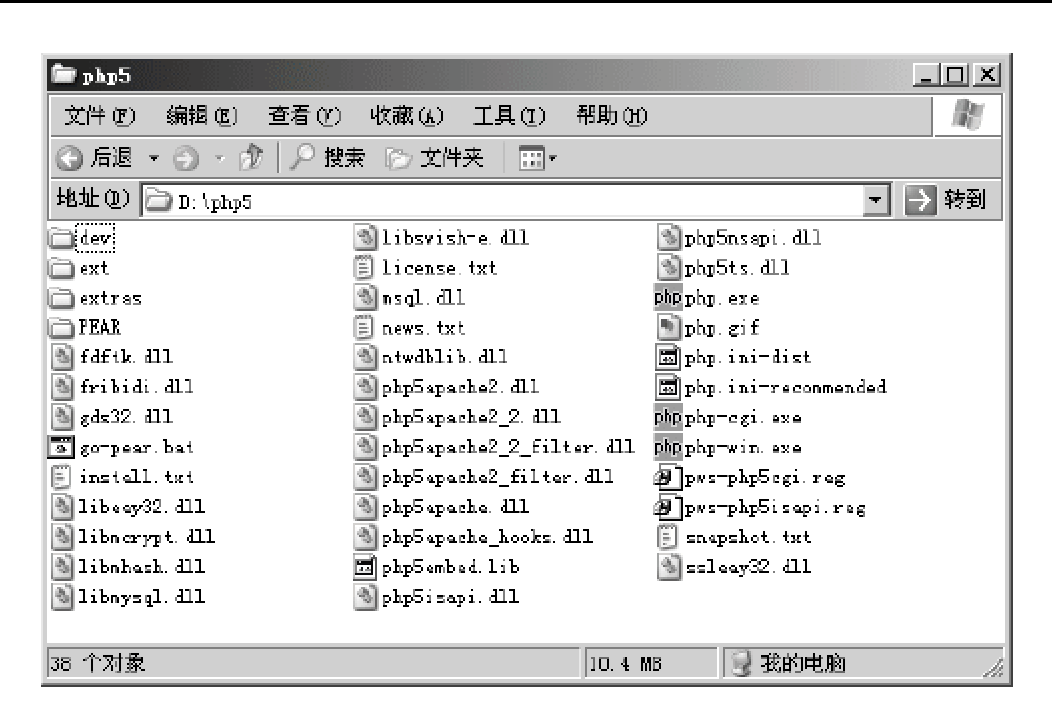


图 2.10　PHP 5 的目录结构

（2）将该目录下的所有 dll 文件复制到系统盘 Windows\system32 目录下（Windows 2000 是在 winnt\system32 目录下）。

（3）将文件 php.ini-dist 文件复制到系统盘\Windows 目录下，并重新命名为 php.ini。

（4）打开 php.ini 文件并找到“extension_dir = "./"”这一行，修改为“extension_dir = "d:/php5/ext"”。

（5）找到“;extension=php_mysql.dll”这一行，将前面的分号“;”去掉。这样，PHP 即可支持 MySQL 数据库。

（6）PHP 配置完成以后，重新启动 Apache 服务器。

（7）编写一个 PHP 脚本文件，命名为 phpinfo.php，保存在 Apache 服务器的虚拟目录 D:/htdocs 下。PHP 脚本文件的代码如下：

```
<?php
    phpinfo();        //获取 PHP 的配置信息
?>
```

最后在浏览器的地址栏中输入“http://localhost/phpinfo.php”，如果显示 PHP 的版本相关信息，则说明 PHP 配置成功。

2.3.2　在 Linux 下安装 PHP

视频讲解：光盘\TM\Video\2\在 Linux 下安装 PHP.exe

安装 PHP 5 之前，需要首先查看 libxml 的版本号。如果 libxml 版本号小于 2.5.10，则需要先安装 libxml 高版本。安装 libxml 和 php 5 的步骤如下（如果不需要安装 libxml，则直接跳到 php 5 的安装步骤即可）：

（1）将 libxml 和 php 5 复制到/usr/local/work/目录下，并进入到该目录。

```
mv php-5.2.5.tar.gz libxml2-2.6.26.tar.gz /usr/local/work
cd /usr/local/work
```

（2）分别将 libxml2 和 php 解压。

```
tar xfz libxml2-2.6.62.tar.gz
tar xfz PHP-5.2.5.tar.gz
```

（3）进入到 libxml 目录，建立 makfile，将 libxml 安装到/usr/local/libxml2 下。

```
cd libxml2-2.6.62
./configure –prefix=/usr/local/libxml2
```

（4）编译文件。

```
makefile
```

（5）开始安装。

```
make install
```

（6）libxml2 安装完毕，开始安装 php 5。进入到 php-5.2.5 目录下。

```
cd ../php-5.2.5
```

（7）建立 makefile。

```
./configure –with-apxs2=/usr/local/Apache2/bin/apxs
--with-mysql=/usr/local/mysql
--with-libxml-dir=/usr/local/libxml2
```

（8）开始编译。

```
make
```

（9）开始安装。

```
make install
```

（10）复制 php.ini-dist 或 php.ini-recommended 到/usr/local/lib 目录，并命名为 php.ini。

```
cp php.ini-dist /usr/local/lib/php.ini
```

（11）更改 httpd.conf 文件相关设置，该文件位于/usr/local/Apache2/conf 中。找到该文件中的如下指令行：

```
AddType application/x-gzip .gz .tgz
```

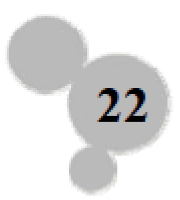
在该指令后加入如下指令：

```
AddType application/x-httpd-php .php
```

重新启动 Apache，并在 Apache 主目录下建立文件 phpinfo.php。

```
<?php
phpinfo();
?>
```

在 Mozilla 浏览器的地址栏中输入“http://localhost/phpinfo.php”，如果出现如图 2.11 所示的页面，说明 PHP 安装成功。



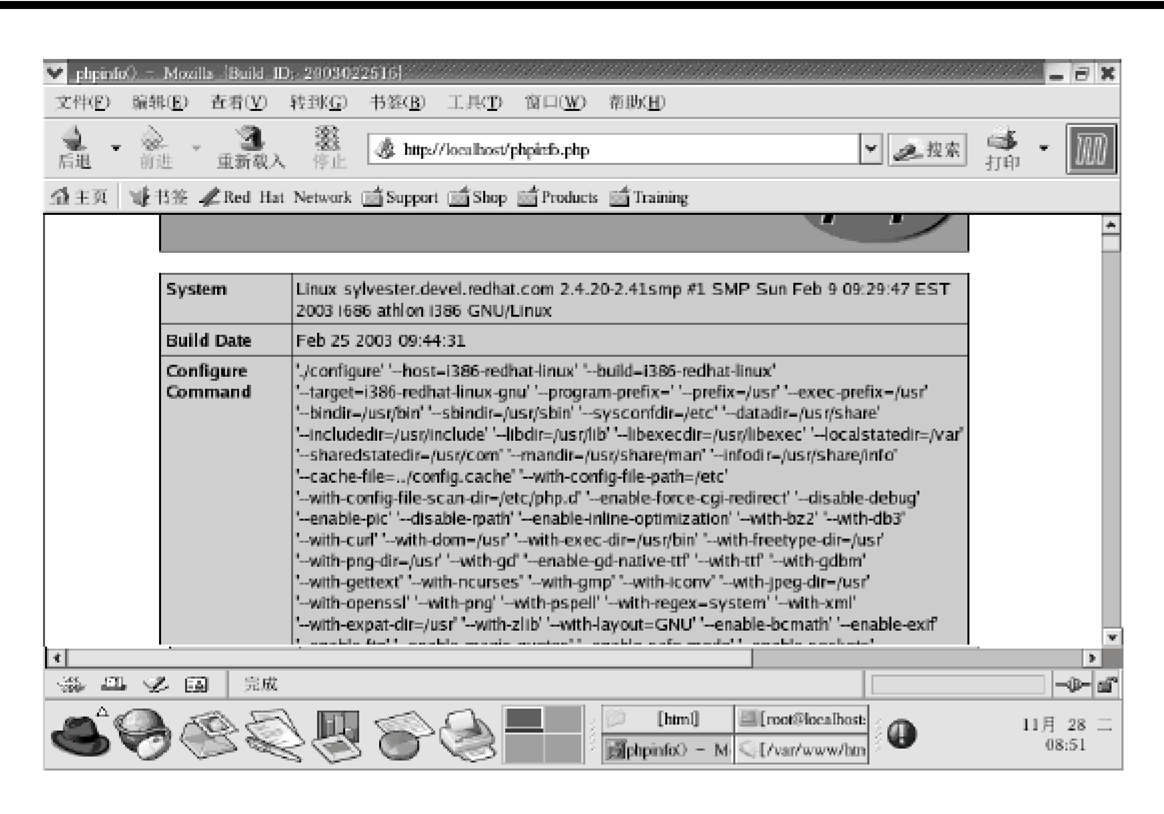


图 2.11　phpinfo 信息

2.4　MySQL 服务器的安装和配置

2.4.1　在 Windows 下安装 MySQL 服务器

视频讲解：光盘\TM\Video\2\在 Windows 下安装 MySQL 服务器.exe

MySQL 是一款广受欢迎的数据库，由于开源所以市场占有率高，备受 PHP 开发者的青睐，一直被认为是 PHP 的最佳搭档。这是因为 MySQL 不仅是完全网络化的跨平台关系型数据库系统，也是具有客户机/服务器体系结构的分布式数据库管理系统。它具有功能性强、使用简捷、管理方便、运行速度快、版本升级快、安全性高等优点，而且 MySQL 数据库完全免费，从官方网站 http://www.mysql.com 即可免费下载到最新版本的 MySQL 安装包 mysql-essential-5.0.51-win32.msi。

说明： *在学习 MySQL 数据库之前，读者几乎不会接触到数据库知识，也使用不到 MySQL 数据库，因此读者可以暂时不用安装 MySQL 而专心学习 PHP 知识，以免在开始学习 MySQL 数据库时由于时间间隔过长，而忘记当初安装时设置的用户名和密码。*

在 Windows 下安装和配置 MySQL 服务器的操作步骤如下。

（1）双击 MySQL 安装文件 mysql-essential-5.0.51-win32.msi，进入欢迎页面。单击 Next 按钮，进入到 Setup Type 页面。

（2）Setup Type 页面中包含 3 个安装选项，第一项是典型安装，第二项是全部安装。这两个安装的路径不能改变，默认是 E:\Program Files\MySQL\MySQL Server 5.0\（E 盘为系统盘）。第三项是自定义安装，允许用户自定义选择安装组件和安装路径。这里选中 Custom 单选按钮。Setup Type 页面的设置如图 2.12 所示。

（3）单击 Next 按钮进入到 Custom Setup 页面。选择需要安装的组件，并单击 Change 按钮来选择要安装的目录。Custom Setup 页面的设置如图 2.13 所示。选择完毕后单击 Next 按钮进入到准备安装页面。

（4）在准备安装页面中显示了用户所选择的安装类型（type）、路径等信息。如果发现前面的选项设置有误，可以单击 Back 按钮返回到上一个页面重新选择；如果正确，则单击 Install 按钮开始安装文件。

（5）文件安装完成后，会出现一些关于 MySQL 的功能和版本的介绍。连续单击 Next 按钮，将会

进入 MySQL 服务器配置页面，如图 2.14 所示。

（6）该页面有两个选项：详细配置（默认）和标准配置。这里保持默认设置。单击 Next 按钮，进入到服务器运行模式页面，如图 2.15 所示。

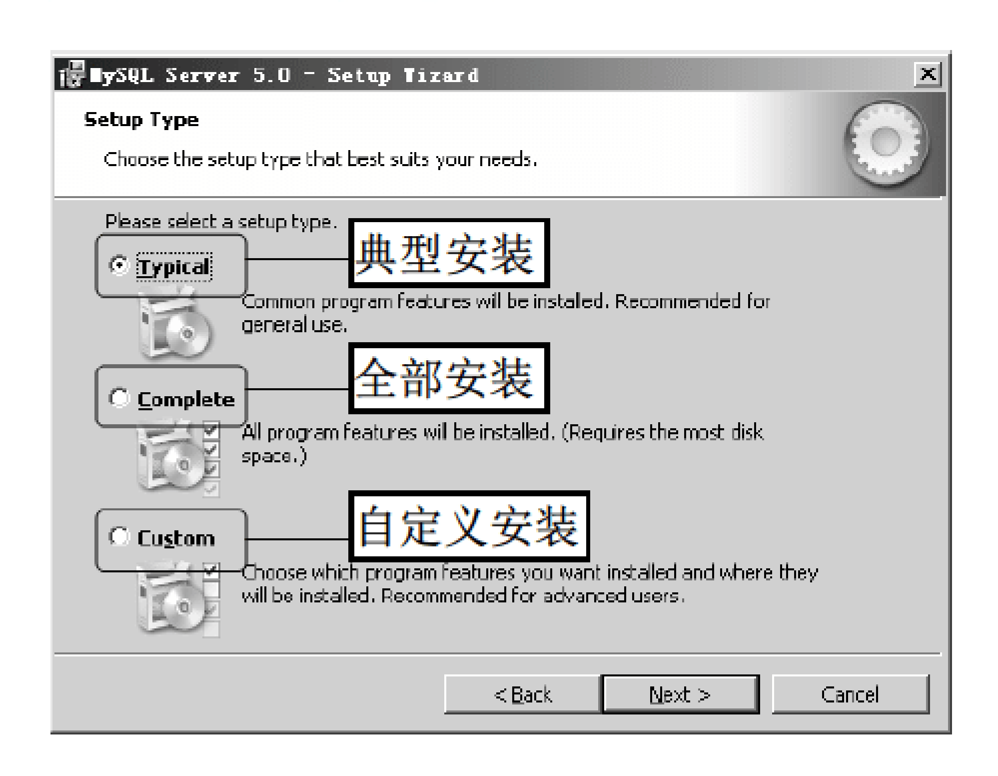


图 2.12　Setup Type 页面

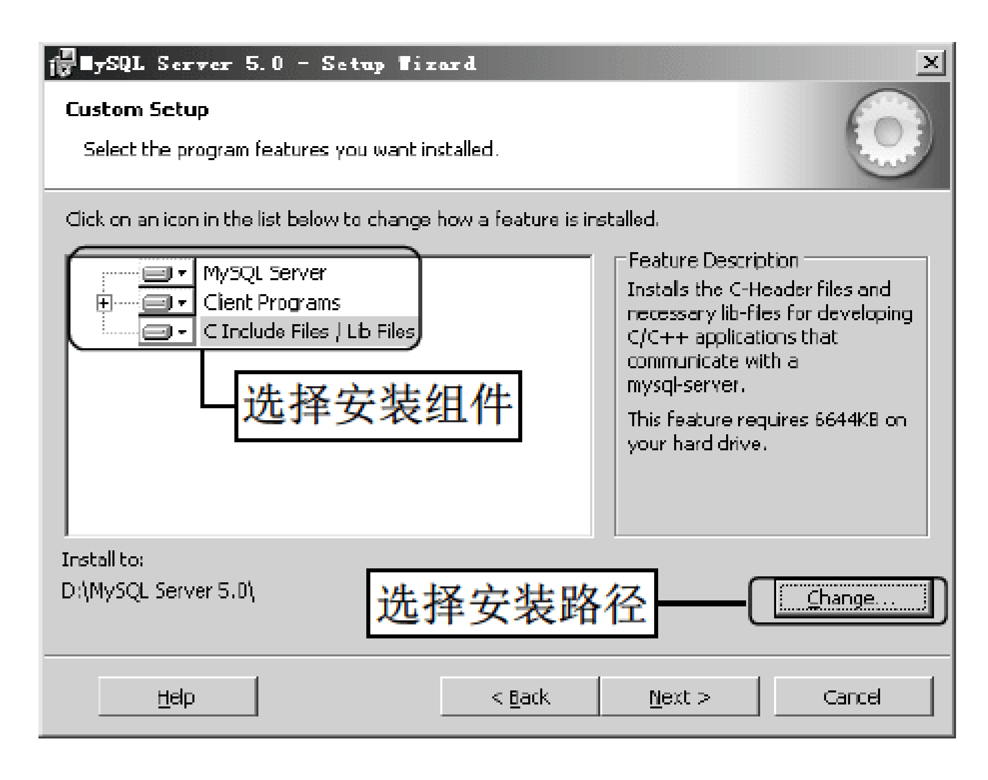


图 2.13　Custom Setup 页面

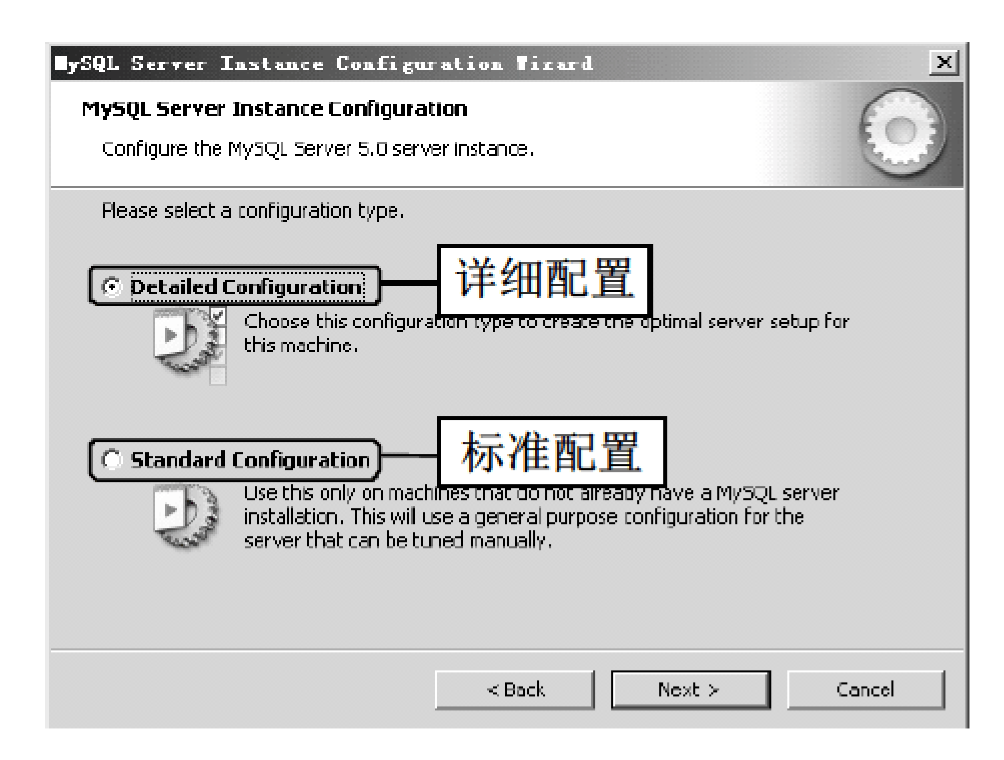


图 2.14　MySQL 服务器配置页面

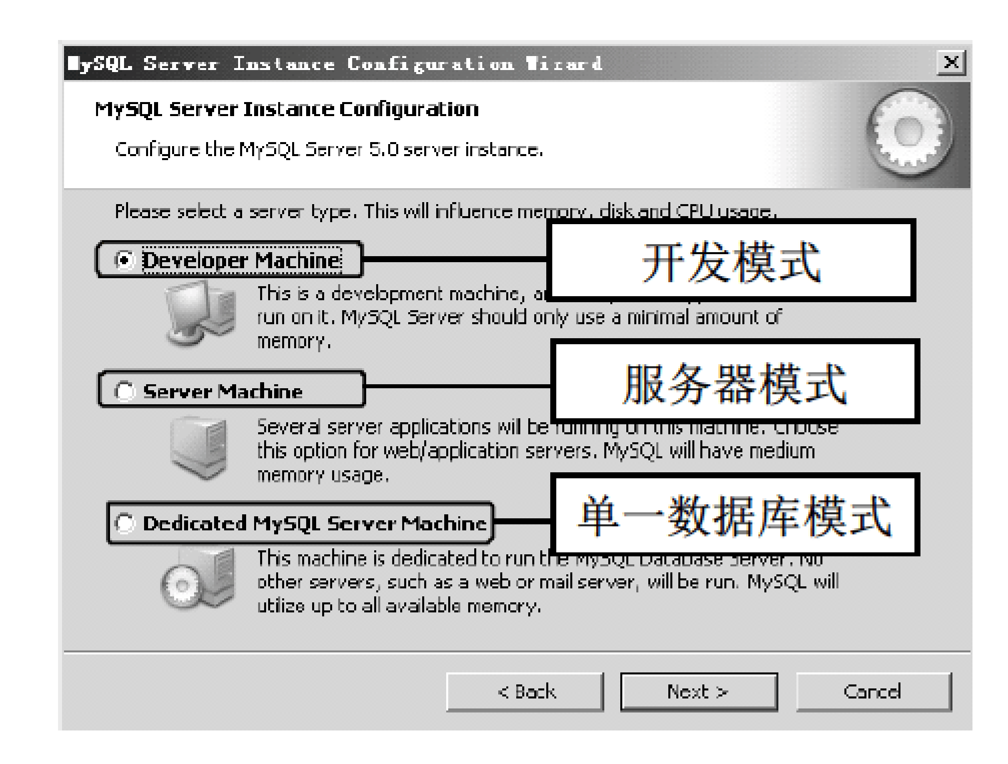


图 2.15　选择服务器运行模式

（7）该页面中有 3 个选项，这里选择第一个默认项即可（即开发模式，MySQL 服务器占用最小的内存空间，作为本地测试使用完全足够）。选择完毕后，单击 Next 按钮进入选择数据库类型页面。

（8）本页面有两种数据库类型的选项，第一项是支持 MyISAM、InnoDB 等多种类型库的数据系统；第二项是只支持其中一种类型库。这里选择默认的第一项：Multifunctional Database，支持多种类型库。单击 Next 按钮，如图 2.16 所示。

（9）进入为 InnoDB 数据文件选择路径页面，这里选择 D 盘下的 MySQL Datafiles 目录。选取分区时要注意所选择分区的剩余空间大小。选择后的页面如图 2.17 所示。单击 Next 按钮。

（10）进入选择同时连接服务器的最大值的页面，这里可以选择默认的第一项，或者选择第三项自定义连接。第二项的最大连接数为 500。选择后的页面如图 2.18 所示。单击 Next 按钮。

（11）进入 MySQL 服务器的端口设置页面，默认 3306 即可。选取完毕后单击 Next 按钮。

（12）进入选择 MySQL 的默认字符集页面。这里选择 GB2312 编码类型。单击 Next 按钮。

（13）进入选择 MySQL 服务器是否自动运行页面。如果要在 Windows 环境变量 path 中加入 MySQL 执行路径，那么需要选中 Include Bin Directory in Windows PATH 复选框。页面设置如图 2.19 所示。单击 Next 按钮，进入到权限设置页面。



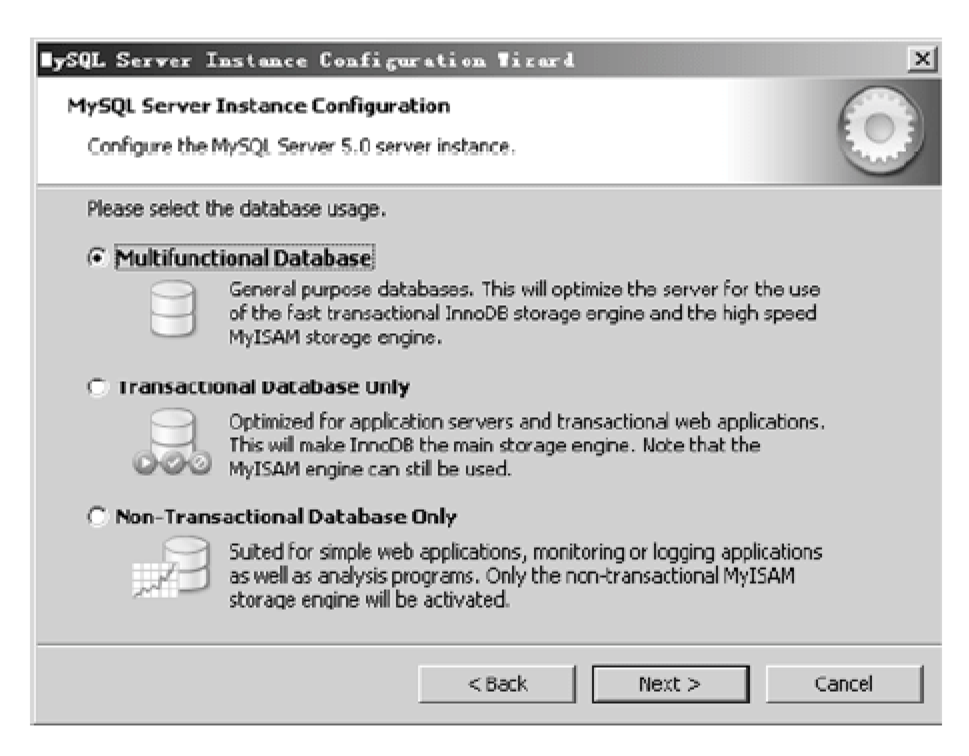


图 2.16　选择数据库类型

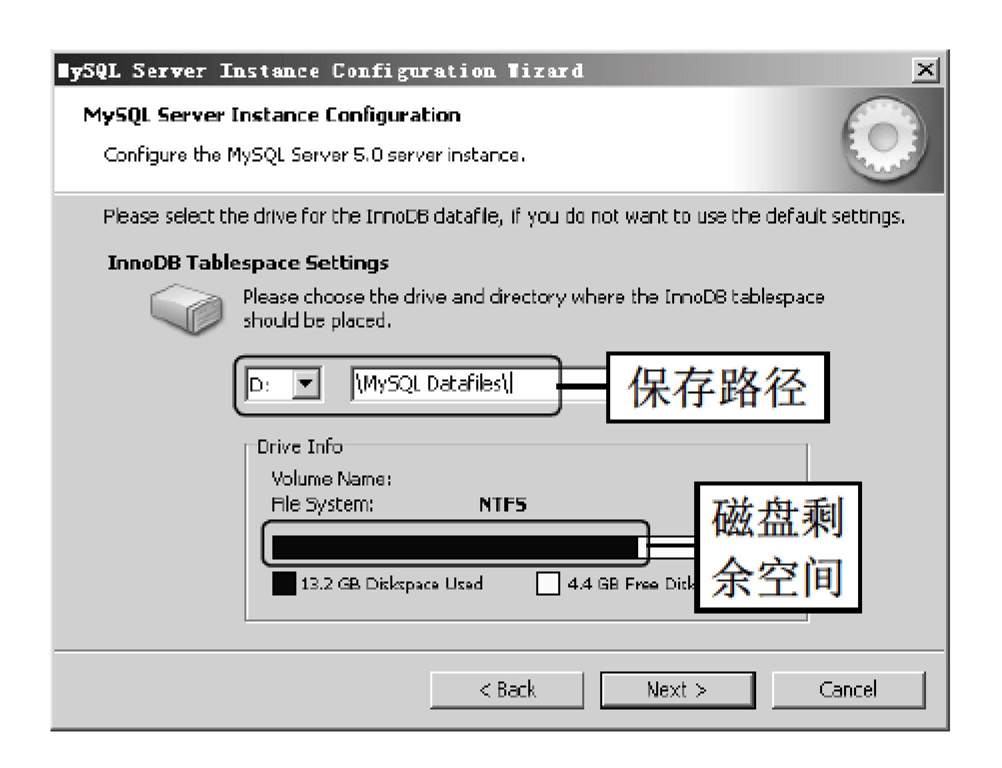


图 2.17　为 InnoDB 数据文件设置路径

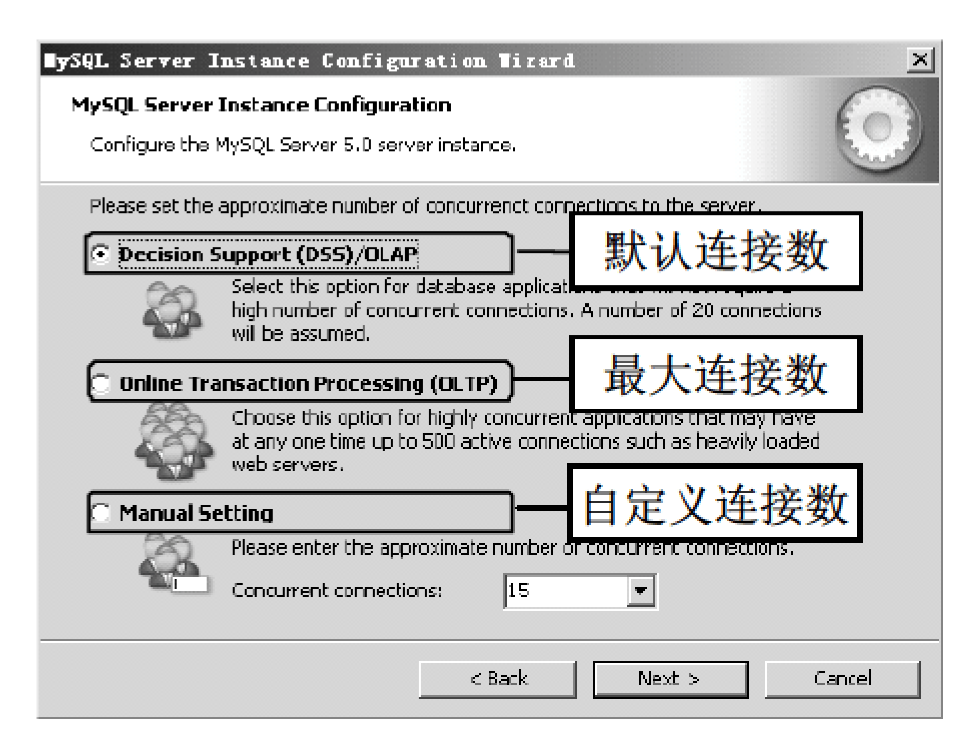


图 2.18　选择同时连接服务器的最大值



图 2.19　选择 MySQL 服务器的启动方式

（14）在该页面中可以设置用户登录密码（本书中所有涉及数据库的实例的密码都为 root，所以这里建议也设置为 root，以方便所有 MySQL 数据库实例的运行），在设置密码的下面有一行文本，询问是否允许 root 用户远程登录数据库。如果选中最下面的复选框，则创建一个允许任何人访问数据库的账号，这里不建议选中。页面设置如图 2.20 所示。

（15）单击 Next 按钮，进入准备执行页面。如果配置没有问题，单击 Execute 按钮开始执行操作，如图 2.21 所示。

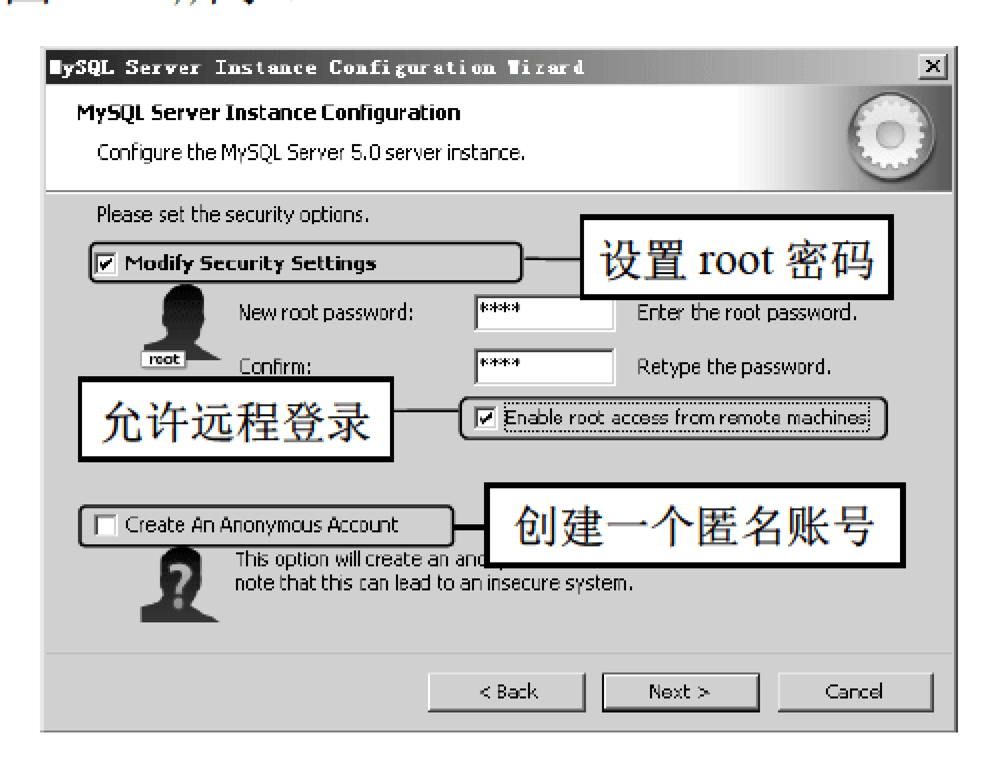


图 2.20　权限设置页面

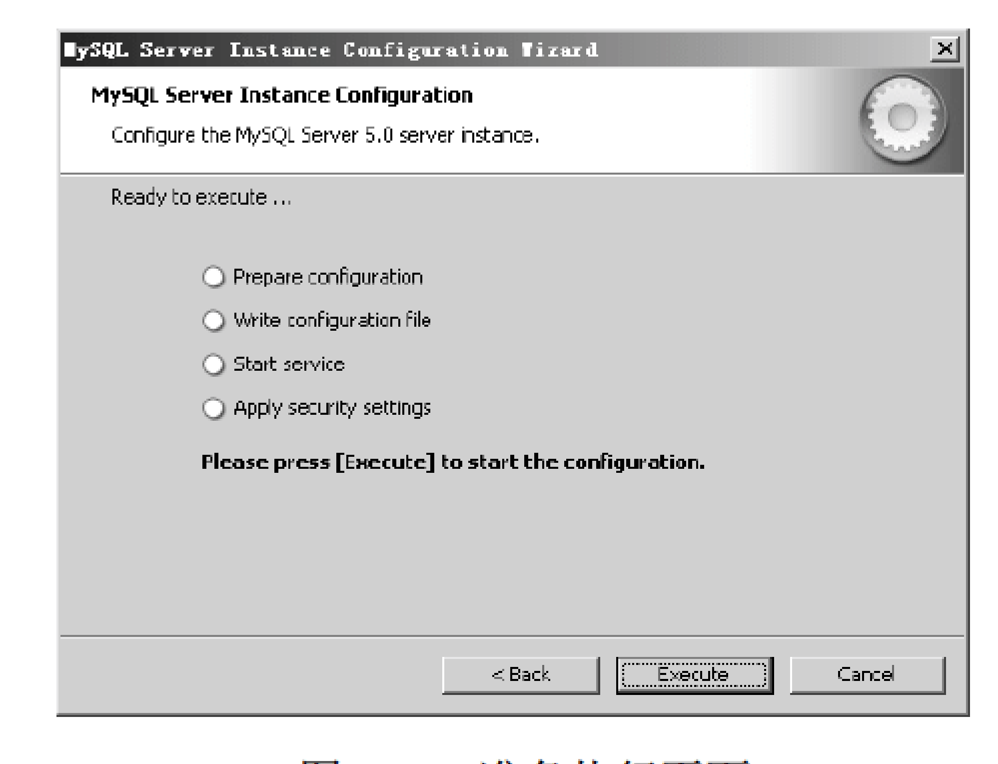


图 2.21　准备执行页面

（16）安装完成后，单击 Finish 按钮完成 MySQL 服务器的安装。



2.4.2　在 Linux 下安装 MySQL 服务器

视频讲解：光盘\TM\Video\2\在 Linux 下安装 MySQL 服务器.exe

在 Linux 系统下安装 MySQL 服务器需要到官方网站 http://www.mysql.com 下载 Linux 下 MySQL 的安装包 mysql-5.0.51a-Linux-i686.tar.gz。

在 Linux 下安装和配置 MySQL 服务器的操作步骤如下。

（1）将下载的 mysql-5.0.51a-Linux-i686.tar.gz 文件复制到/usr/local/work 目录下，创建 MySQL 账号，并加入组群。

```
groupadd mysql
useradd –g mysql mysql
```

（2）进入到 MySQL 的安装目录，将其解压（如目录为/usr/local/mysql）。

```
cd /usr/local/mysql
tar xfz /usr/local/work/mysql-5.0.51a-Linux-i686.tar.gz
```

（3）考虑到 MySQL 数据库升级的需要，所以通常以链接的方式建立/usr/local/mysql 目录。

```
ln –s mysql-5.0.51a-Linux-i686.tar.gz mysql
```

（4）进入到 MySQL 目录，在/usr/local/mysql/data 中建立 MySQL 数据库。

```
cd mysql
scripts/mysql_install_db –user=mysql
```

（5）修改文件权限。

```
chown –R root
chown –R mysql data
chgrp –R mysql
```

（6）到此 MySQL 安装成功。用户可以通过在终端中输入命令启动 MySQL 服务。

```
/usr/local/mysql/bin/mysqld_safe –user=mysql &
```

启动后输入命令，进入 MySQL。

```
/user/local/mysql/bin/mysql –uroot
```

如果终端页面显示如图 2.22 所示的提示信息，则说明 MySQL 服务器安装成功。

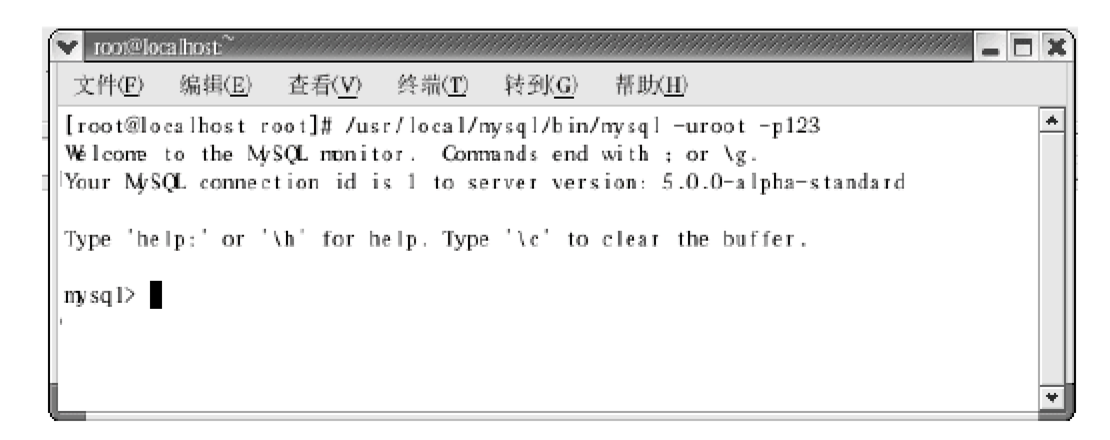
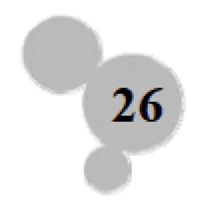


图 2.22　测试 MySQL 是否安装成功



2.5　初学者——应用组合包快速搭建 PHP 环境

组合包，就是将 Apache、PHP 和 MySQL 等服务器软件和工具安装配置完成后打包处理。开发人员只要将已配置的套件解压到本地硬盘中即可使用，无须再另行配置。组合包实现了 PHP 开发环境的快速搭建。对于刚刚起步学习 PHP 的程序员，建议采用此方法搭建 PHP 的运行环境。组合包安装简单、速度较快、运行稳定，使用户能将精力更好地集中到 PHP 的学习中。

目前网上流行的组合包有十几种，安装基本都是大同小异。这里推荐 3 种组合包：EasyPHP、AppServ、XAMPP。

建议新手使用 EasyPHP 或 AppServ，两者都是 Apache+MySQL+PHP 开发环境的。而 XAMPP 则相对要复杂一些。不仅可以切换 PHP 4 和 PHP 5，还集成了 perl 开发环境、第三方扩展库等，并且对开发平台进行了优化和整理。如果对 PHP 的开发环境已经有了一定的了解，则推荐使用 XAMPP。

☑　EasyPHP 下载地址为 http://www.easyphp.org/。
☑　AppServ 下载地址为 http://www.appservnetwork.com/。
☑　XAMPP 下载地址为 http://www.Apachefriends.org/。

注意：要安装这些组合包，必须保证系统中未安装 Apache、PHP 和 MySQL。否则，要先将这些软件卸载后，再开始安装组合包。组合包的安装很简单，只要将程序解压或安装到指定目录后就可以直接使用。

本节以 AppServ 组合包的 AppServ-win32-2.5.7 版本为例，重点讲解该组合包的安装和使用方法。

AppServ 是 PHP 网页架站工具组合包，是将网络上免费的架站资源重新包装成单一的安装程序。它提供了简易、快速的 PHP 运行环境的搭建机制，读者只需按照普通应用软件的安装方式就可以完成 Apache+MySQL+PHP+phpMyAdmin 的安装与配置工作。可以说，AppServ 是初学者的首选。

安装 AppServ 之前应从官方网站 http://www.appservnetwork.com 下载 AppServ-win32-2.5.7.exe 安装程序。

视频讲解：光盘\TM\Video\2\应用组合包快速搭建 PHP 环境.exe

在 Windows 下应用 AppServ 组合包快速搭建 PHP 开发环境的操作步骤如下。

（1）双击 AppServ-win32-2.5.7.exe 文件，打开如图 2.23 所示的 AppServ 启动页面。

（2）单击 Next 按钮，打开如图 2.24 所示的 AppServ 安装协议页面。

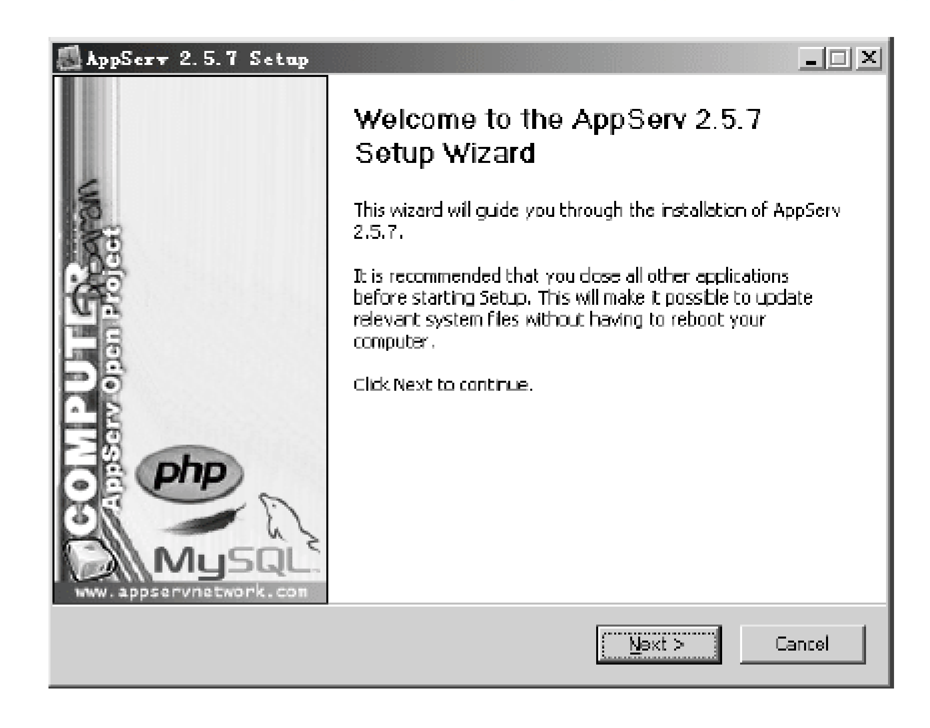


图 2.23　AppServ 启动页面

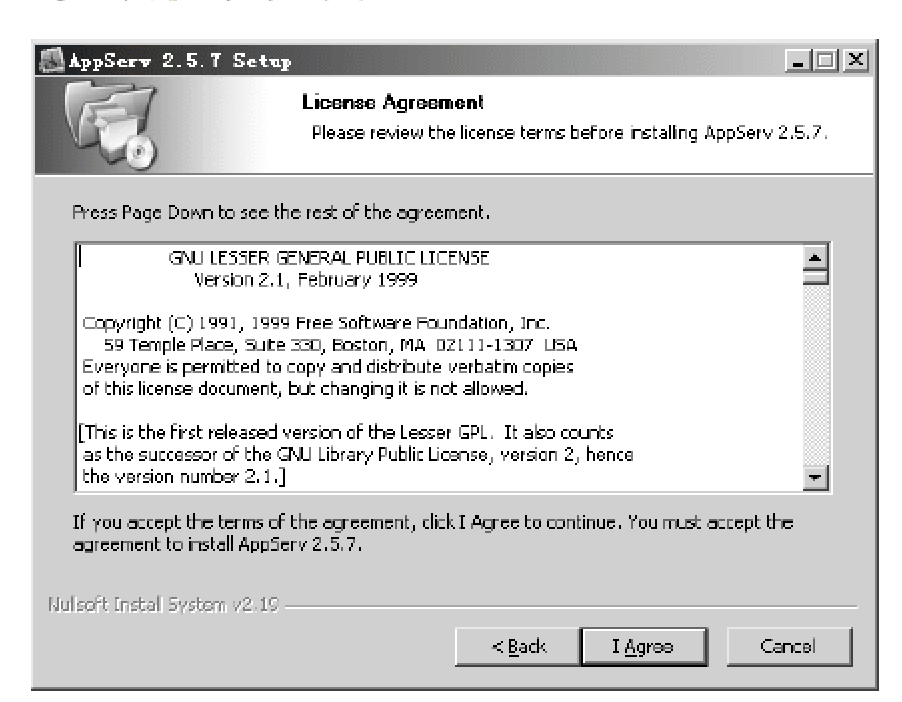


图 2.24　AppServ 安装协议页面

（3）单击 I Agree 按钮将打开如图 2.25 所示的页面。在该页面中可以设置 AppServ 的安装路径（默认安装路径一般为 C:\AppServ），AppServ 安装完成后 Apache、MySQL、PHP 都将以子目录的形式存储到该目录下。

（4）单击 Next 按钮将打开如图 2.26 所示的页面。在该页面中可以选择要安装的程序和组件（默认为全选状态）。

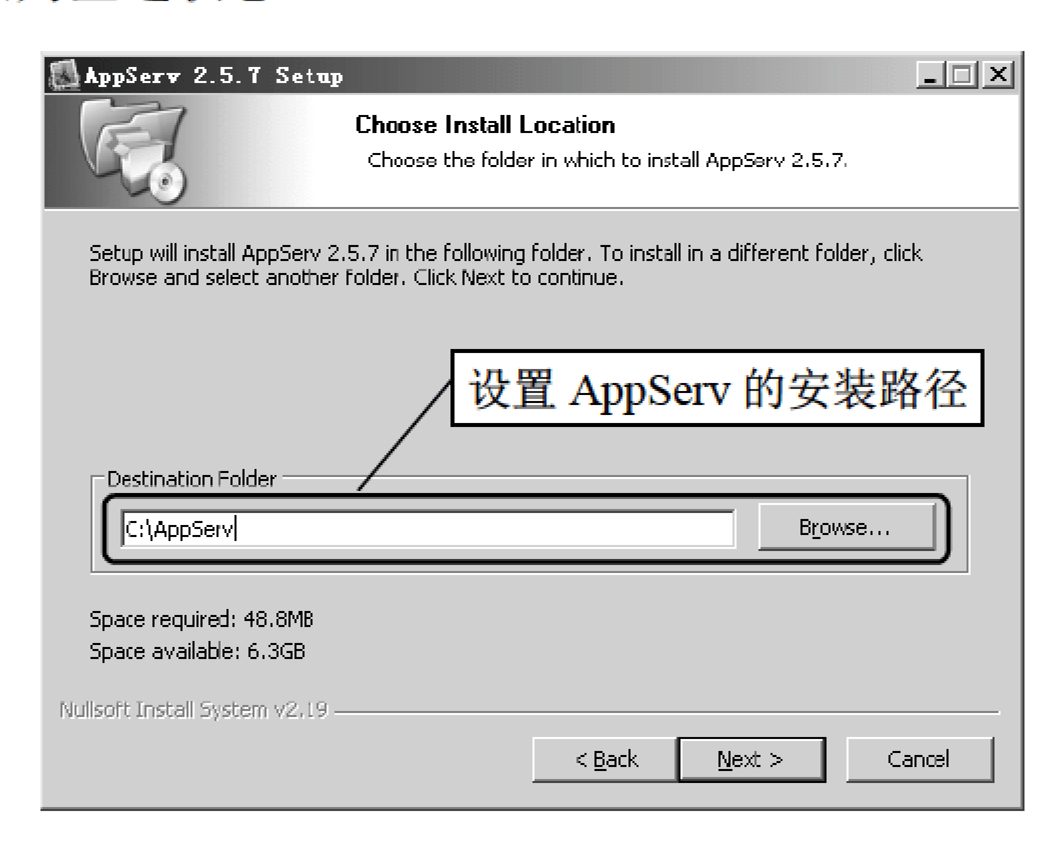


图 2.25　选择 AppServ 安装路径

图 2.26　AppServ 安装选项

（5）单击 Next 按钮，打开如图 2.27 所示的页面。该页面主要用于设置 Apache 的端口号。

（6）单击 Next 按钮，打开如图 2.28 所示的页面。该页面主要用于对 MySQL 数据库的 root 用户的登录密码及字符集进行设置，这里将字符集设置为 GB2312 Simplified Chinese，表示 MySQL 数据库的字符集将采用简体中文形式。



图 2.27　设置 Apache 端口号

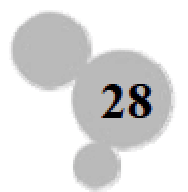


图 2.28　设置 MySQL 登录密码及字符集

（7）单击 Install 按钮后开始安装，如图 2.29 所示。

（8）安装完成后可以在开始菜单的 AppServ 相关操作列表中启动 Apache 及 MySQL 服务，如图 2.30 所示。

（9）安装好 AppServ 之后，整个目录默认为 C:\AppServ，此目录下包含 4 个子目录，如图 2.31 所示，用户可以将所有网页文件存放到 www 目录下。



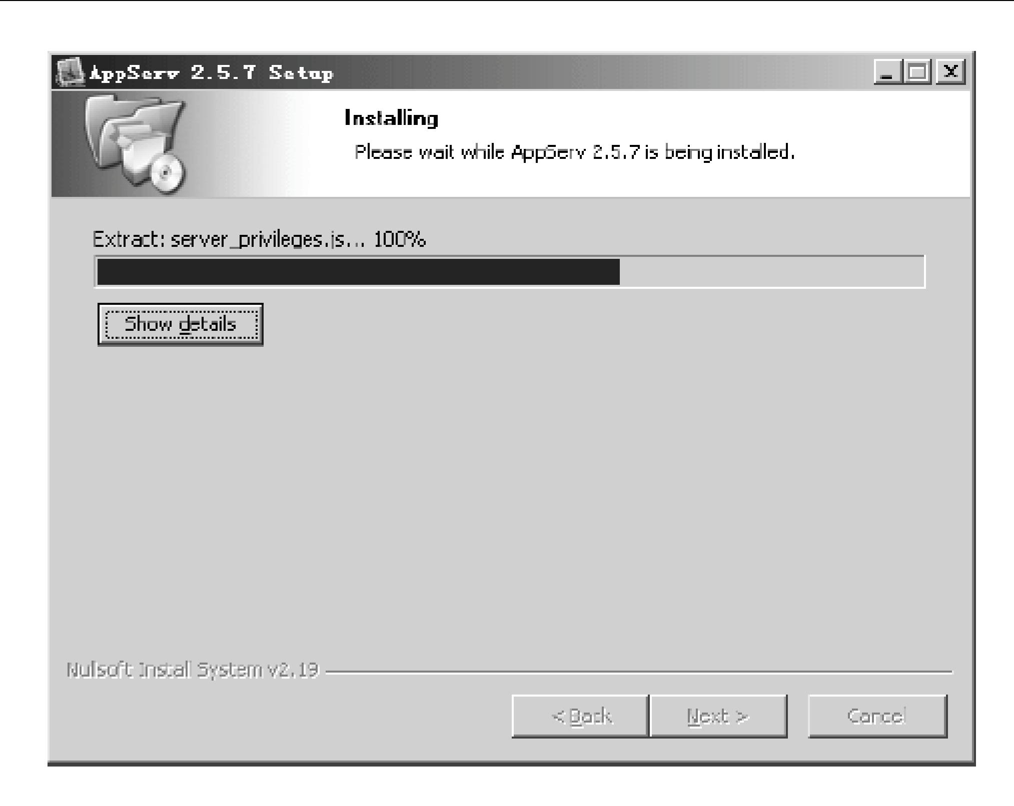


图 2.29　AppServ 安装进度页面

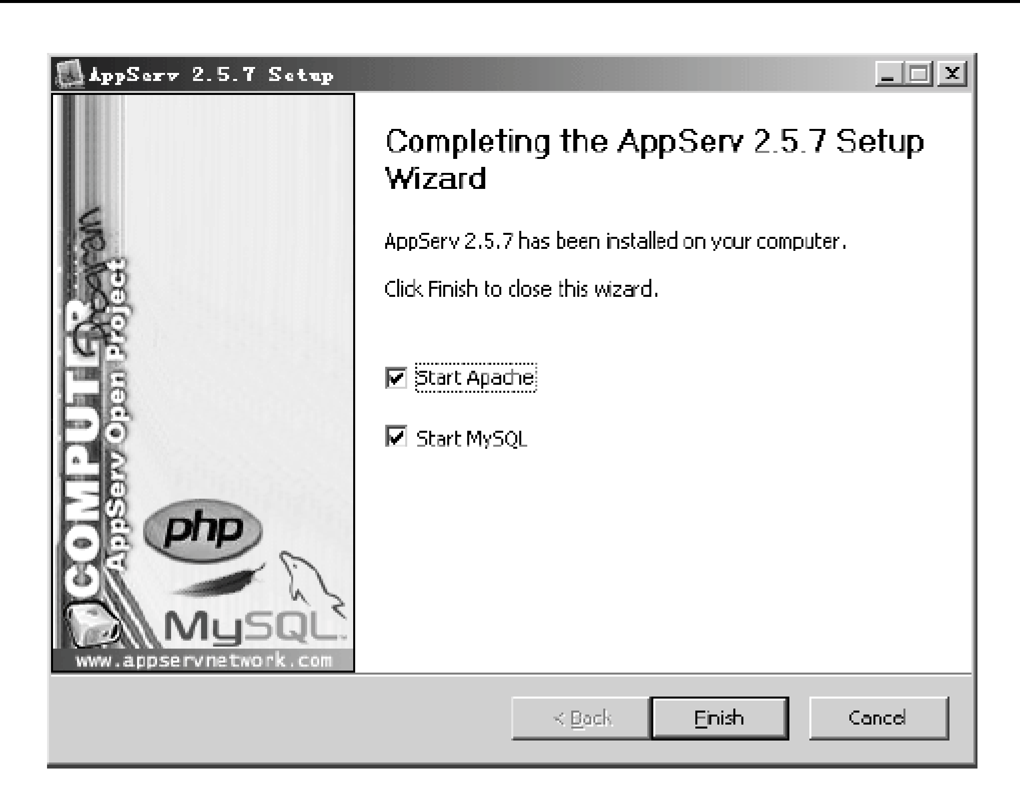


图 2.30　AppServ 安装完成页面

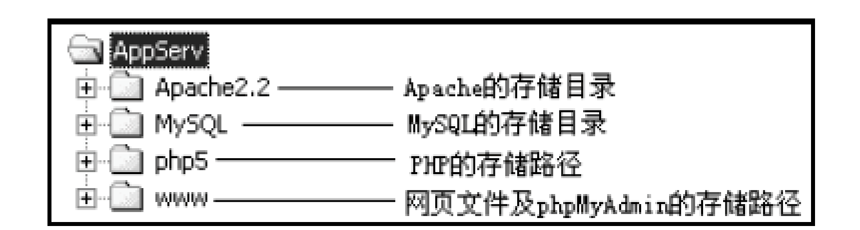


图 2.31　AppServ 目录结构

（10）打开浏览器，在 IE 浏览器的地址栏中输入“http://localhost/”或者“http://127.0.0.1/”，如果打开如图 2.32 所示的页面，则说明 AppServ 安装成功。

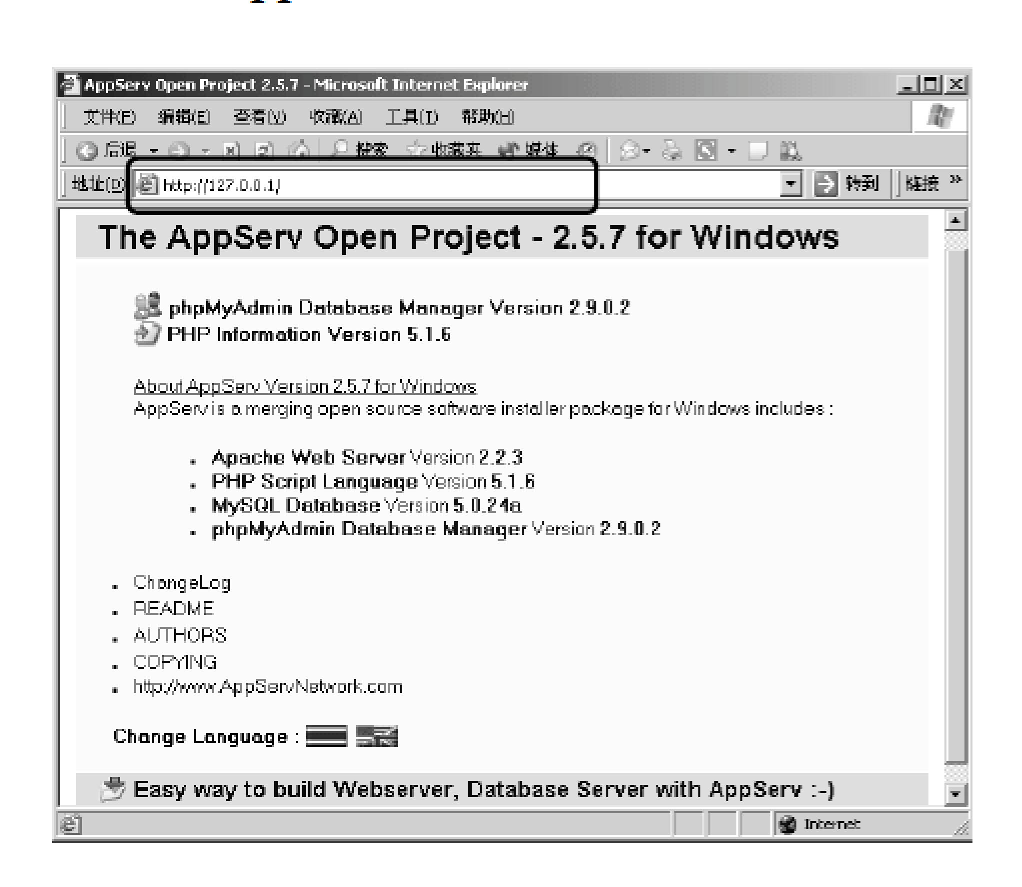


图 2.32　AppServ 测试页

2.6　第一个 PHP 程序

视频讲解：光盘\TM\Video\2\第一个 PHP 程序.exe

2.6.1　使用 Macromedia Dreamweaver 8 编写源程序

服务器环境配置完成后，接下来应用 Macromedia Dreamweaver 8 开发工具来编写第一个 PHP 程序。

例 2.1 编写第一个 PHP 程序的目的是熟悉 PHP 的书写规则和 Macromedia Dreamweaver 8 工具的基本使用方法。本实例很简单，就像大多数语言一样，输出一段欢迎信息。开发步骤如下：**（实例位置：光盘\TM\Instances\2\1）**

（1）启动 Macromedia Dreamweaver 8 编辑器，选择“文件”/“新建”命令，打开“新建文档”对话框，在“常规”选项卡中选择“动态页”类别，在“动态页”列表框中选择 PHP 选项，如图 2.33 所示。

（2）单击“创建”按钮，即可成功创建一个动态的 PHP 页面。对新创建的页面，可以在“代码”视图中编辑 PHP 代码，也可以使用“设计”视图查看 HTML 效果。这里使用“代码”视图，并给该页面标题命名。如图 2.34 所示，标题显示的位置是在浏览器的左上角，稍后运行时就能看到效果。



图 2.33 “新建文档”对话框

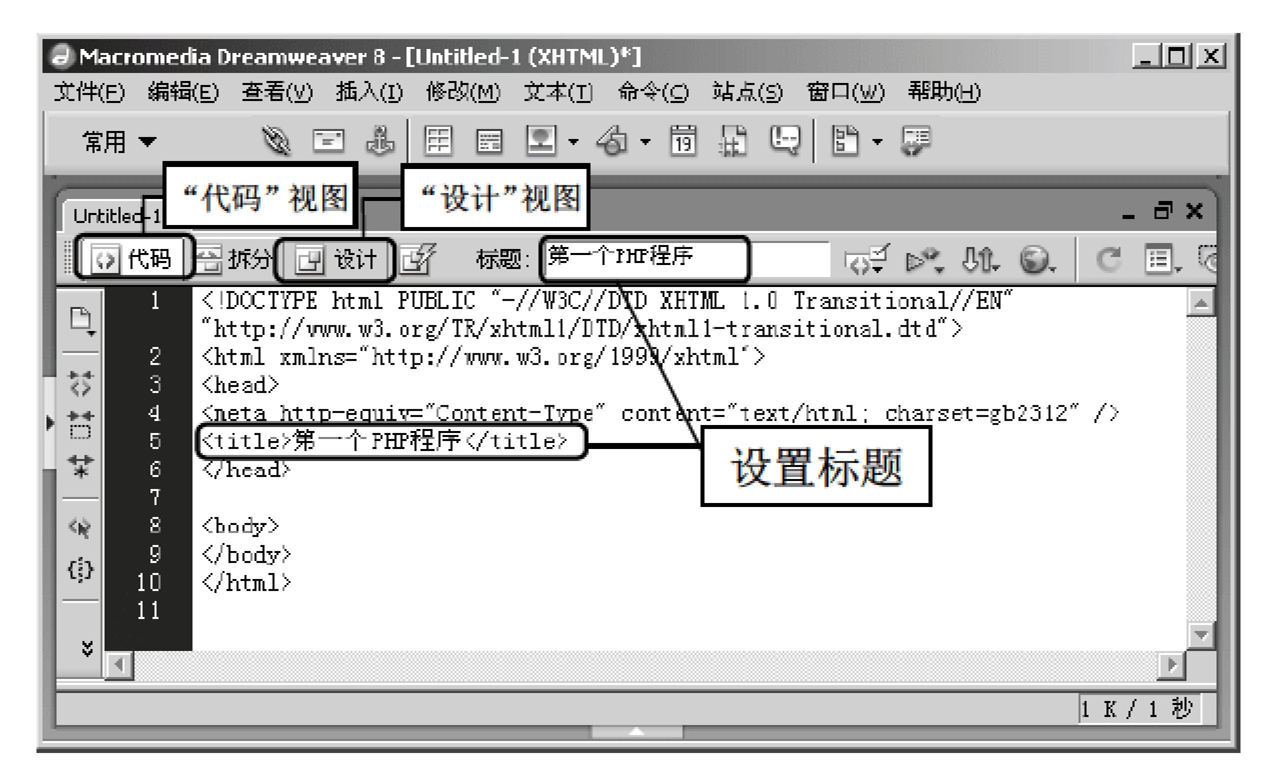


图 2.34 命名标题

（3）编写 PHP 代码。在<body>…</body>标签对中间开始编写 PHP 代码段，代码如下。

```
<?php
    echo  "欢迎进入 PHP 的世界！！";
?>
```

代码讲解：

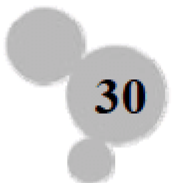
☑ “<?php”和“?>”是 PHP 的标记符。在这对标记符中的所有代码都被当作 PHP 代码来处理。除了这种表示方法外，PHP 还可以使用 ASP 风格的“<%”和 SGML 风格的“<?…?>”等，在第 3 章中将会详细介绍。

☑ echo 语句是 PHP 中最常用的语句，主要用于将一个或多个字符串输出至网页，和 ASP 中的 response.write 以及 JSP 中的 out.print 是一个意思，就是将紧跟在后面的字符串或者变量值显示在页面中。每行代码都以分号“;”结尾。

注意： 在使用 echo 语句时，应该时刻牢记语句结束处的分号。在 PHP 中，分号是用来分隔语句的。丢失这个“;”是读者经常会犯的语法错误，同样也是读者在进行程序的错误处理时，首先应该注意的地方。

在 Macromedia Dreamweaver 8 中输入的 PHP 脚本程序如图 2.35 所示。

（4）将 PHP 动态页保存到服务器指定的目录以便解析。本章中服务器指定的目录为 D:\appserv\



www\，则将本页保存到 D:\appserv\www\tm\sl\2\1\路径下，命名为 index.php。

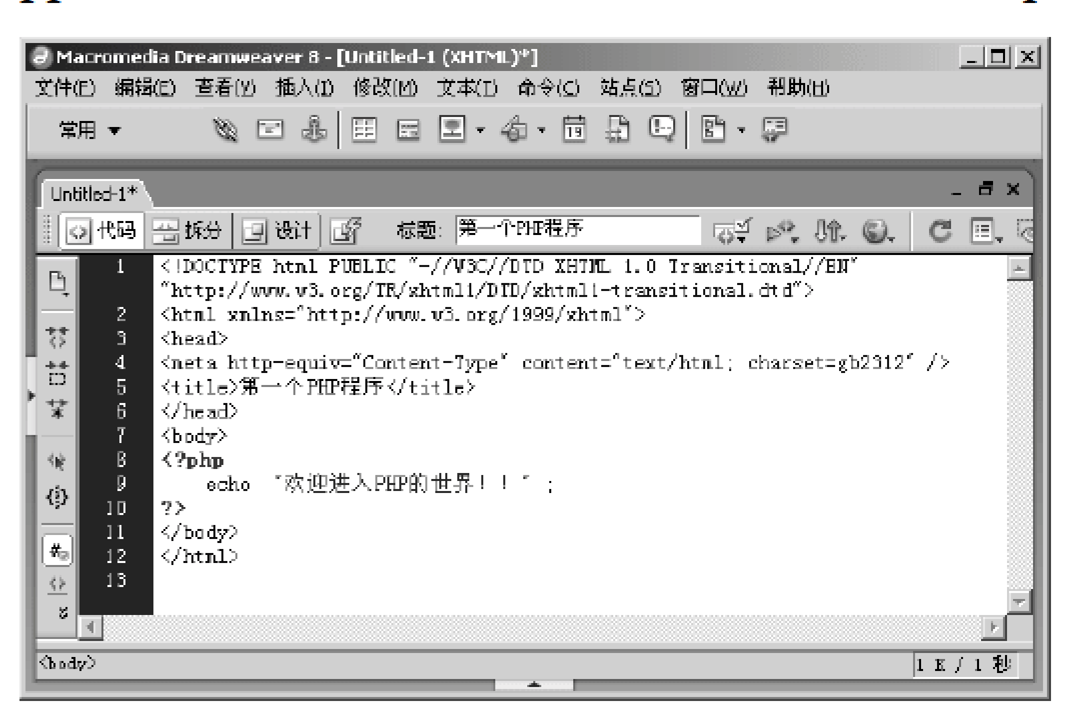


图 2.35　在开发工具中输入 PHP 脚本程序

2.6.2　发布和运行 PHP 程序

打开 IE 浏览器页面，在地址栏中输入“http://127.0.0.1/tm/sl/2/1/index.php”，按 Enter 键后即可查看 index.php 页的执行结果，如图 2.36 所示。



图 2.36　PHP 页面运行结果

2.7　环境安装常见问题

2.7.1　Apache 安装常见问题

1．解决 Apache 服务器端口冲突

IIS 服务器、迅雷的默认端口号为 80，同 Apache 服务器默认端口号相同。两者由于采用了相同的端口号 80，因此，在运行网页时就会发生冲突。

如果用户安装了 IIS 服务器，就需要修改 IIS 的默认端口，否则将导致 Apache 服务器无法正常工作。更改 IIS 的默认侦听端口 80，可以在 IIS 的管理器中进行设置，也可以停止 IIS 的服务。

如果用户安装并开启了迅雷软件，就需要关闭该软件，否则端口冲突将会导致运行 PHP 网页程序时出错。

用户也可以在安装 Apache 服务器时将默认的端口号进行更改，从而解决两个服务器或与其他软件共用一个端口号而产生冲突的问题。

2．更改 Apache 服务器默认存储的文件路径

Apache 服务器的核心配置文件是 httpd.conf，存放路径为“Apache 的安装路径\conf\”，用记事本

```
    print"<table width='750' border='1' cellpadding='1' cellspacing='1' bordercolor='#FFFFFF' bgcolor='#99CC00'><tr><td width='131' align='center' bgcolor='#FFFFFF'><span class='STYLE_5'>标题</span></td><td width='258' align='center' bgcolor='#FFFFFF'><span class='STYLE_5'>链接地址</span></td><td width='343' align='center' bgcolor='#FFFFFF'><span class='STYLE_5'>内容摘要</span></td></tr>";
    for($i=0;$i<$node_record->length;$i++){                                //根据标签的数量执行 for 循环
        $k=0;
        foreach($node_record->item($i)->childNodes as $articles){         //通过 foreach 语句输出标签中的数据
            $field[$k]=iconv("UTF-8","GB2312",$articles->textContent);
              $k++;
    }
    print"<tr><td align='left' bgcolor='#FFFFFF'>";
    print"$field[0]";
    print"</td><td align='left' bgcolor='#FFFFFF'>";
   print"$field[1]";
    print"</td><td align='left' bgcolor='#FFFFFF'>";
    print"$field[2]";
    print"</td></tr>";
    }
    print"</table>";
}
}
?>
```

（2）实例化 Message_XML 类，并通过实例化的对象$hawkXML 调用 show_message()方法，输出存储在 XML 文件中的内容，关键代码如下。

```
<?php
  $hawkXML=new Message_XML;
  $Action=$_GET['Action'];
  switch($Action){
        case "show_message":
              $hawkXML->show_message();
         break;
  }
  ?>
```

运行结果如图 22.10 所示。



图 22.10　通过 DOM 读取 XML 文档中的数据

PHP 中的 DOM 函数库十分庞大，这里只是对创建和读取 XML 文档所使用的方法进行了介绍。感兴趣的读者可以参考 XML 和 PHP 的官方手册来了解更多关于 DOM 的知识。

22.5.3　通过 XPath 查询 XML 中的数据

视频讲解：光盘\TM\Video\22\通过 XPath 查询 XML 中的数据.exe

通过 DOM 类库查询 XML 数据主要应用的是 XPath，通过 DOMXPath->query()方法来执行查询特定的节点。DOMXPath->query()方法在 XML 中执行查询的操作，查询符合一些模式语法的特定节点。返回值为一个节点列表对象。语法如下：

```
class DOMXPath {
    DOMNodeList query ( string expression [, DOMNode contextnode] )
}
```

参数 expression 是执行的查询语句，该参数的输出写法有多种，进而可分割执行不同的查询操作。例如：

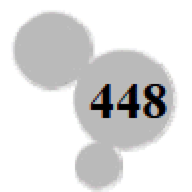
```
/rss/channel/item[position() = 1]/title     返回第一个 item 元素的所有 title
/rss/channel/item/title[@id = '23']         返回所有含有 id 属性且值为 23 的 title
/rss/channel/&folder&/title                 返回所有 articles 元素下面的 title（&folder&代表目录深度）
```

其返回值为，如果成功则返回一个节点列表对象 DOMNodeList，如果失败则返回一个空的节点列表对象 DOMNodeList。

例 22.14　应用 DOMXPath->query()方法，实现对 XML 文档中的数据进行查询操作，查询 XML 文档中标题是“明日科技”的数据，将查询结果输出到文本框中，操作步骤如下：**（实例位置：光盘\TM\Instances\22\14）**

（1）编写一个 Message_XML 类，该类继承于 DomDocument 类，应用该类的 load()方法调用 XML 文件 Rss.xml，在 Message_XML 类中建立 select_message()方法，通过该函数实现查询 XML 文件中数据的操作，代码如下。

```
<?php
    class Message_XML extends DomDocument{                                  //定义类
    private $Root;
    public function __construct() {                                         //构造函数
        parent:: __construct();
        $this->load("Rss.xml");}                                            //加载指定的文件
public function select_message($Action){                                    //创建方法
    $Root   = $this->documentElement;
    $xpath = new DOMXPath($this);                                           //定义 DOMXPath
    $node_Record = $xpath->query("//item[title='$Action']");                //执行查询语句
    for($i=0;$i<$node_Record->length;$i++){                                 //应用 for 循环输出查询结果
        $g=0;
        foreach($node_Record->item($i)->childNodes as $node){//通过 foreach 语句读取返回对象中的数据
            $field[$g]=iconv("UTF-8","GB2312",$node->textContent);          //实现编码格式的转换
            $g++;
```



```
			}
		//输出查询结果
		//省略了输出查询结果的代码
		}
	}
}
?>
```

（2）创建一个操作界面，插入一个表单、文本框，设置提交按钮，然后对 Message_XML 类进行实例化，并应用实例化后的对象$hawkXML 调用 select_message()方法执行查询操作，关键代码如下。

```
<?php
	if($Submit==true){
		$hawkXML = new Message_XML;
		$hawkXML->select_message(iconv("GB2312","UTF-8","$_POST[Action]"));
	}
?>
```

技巧： 在执行查询操作过程中，必须要对提交的字符串进行编码格式的转换，因为在 XML 文件中使用的是 UTF-8 的编码格式，而程序中提交的数据使用的是 GB2312 编码，所以要将提交的数据转换成 UTF-8 的编码格式，否则是不能进行匹配查询的。对编码格式的转换使用的是 iconv()函数。

运行本实例，在文本框中输入关键字“明日科技”，将输入如图 22.11 所示的内容。

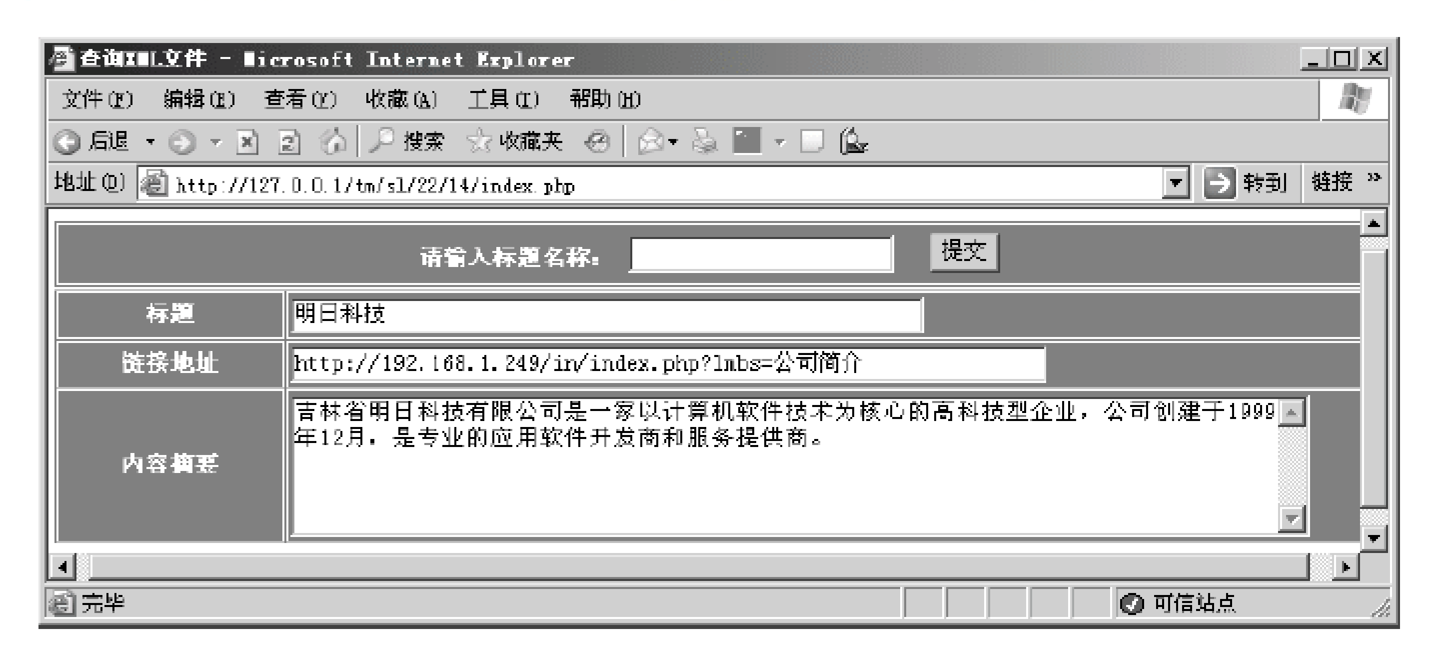


图 22.11　通过 XPath 查询 XML 中的数据

22.6　本 章 小 结

本章首先介绍了 XML 的基础语法，使读者对 XML 有了一个初步的印象；然后讲解了如何在 PHP 中创建一个 XML 文档；接着对 PHP 5 最新的 SimpleXML 类库进行了详细的介绍；最后使用 DOM 对象模型动态创建了一个 XML 文档。

希望读者可以通过本章的概念和实例，初步掌握 PHP 对 XML 文档的操作，为学习后面的 Ajax 技术做好准备。

22.7 实 战 练 习

1．通过 DOM 类库向 XML 中添加数据。**（答案位置：光盘\TM\Instances\22\15）**

2．通过 replaceChild()方法修改 XML 文档。**（答案位置：光盘\TM\Instances\22\16）**

3．通过 removeChild()方法删除 XML 文档。**（答案位置：光盘\TM\Instances\22\17）**

23.1 Ajax 概述

随着 Web 2.0 时代的到来，Ajax 技术产生并逐渐成为主流的 Web 应用。相对于传统的 Web 应用开发，Ajax 运用的是更加先进、更加标准化、更加高效的 Web 开发技术体系。由于 Ajax 是客户端技术，其编译和执行由浏览器完成，所以无论哪种 Web 语言都可以使用 Ajax 技术开发前台应用和页面特效。本章主要介绍 Ajax 技术及如何在 PHP 中应用 Ajax 技术。

23.1.1 什么是 Ajax

Ajax 是 JavaScript、XML、CSS、DOM 等多种已有技术的组合，可以实现客户端的异步请求操作，这样可以实现在不需要刷新页面的情况下与服务器进行通信，从而减少了用户的等待时间。Ajax 是由 Jesse James Garrett 创造的，是 Asynchronous JavaScript And XML 的缩写，即异步 JavaScript 和 XML 技术。可以说，Ajax 是“增强的 JavaScript”，是一种可以调用后台服务器获得数据的客户端 JavaScript 技术，支持更新部分页面的内容而不重载整个页面。

23.1.2 Ajax 的开发模式

传统的 Web 应用模式中，页面中用户的每一次操作都将触发一次返回 Web 服务器的 HTTP 请求，服务器进行相应的处理后，返回一个 HTML 页面给客户端浏览器。Ajax 中，页面中用户的操作将通过 Ajax 引擎与服务器端进行通信，然后将返回结果提交给客户端页面的 Ajax 引擎，再由 Ajax 引擎来决定将这些数据插入到页面的指定位置。一个典型的 Ajax 应用如图 23.1 所示。

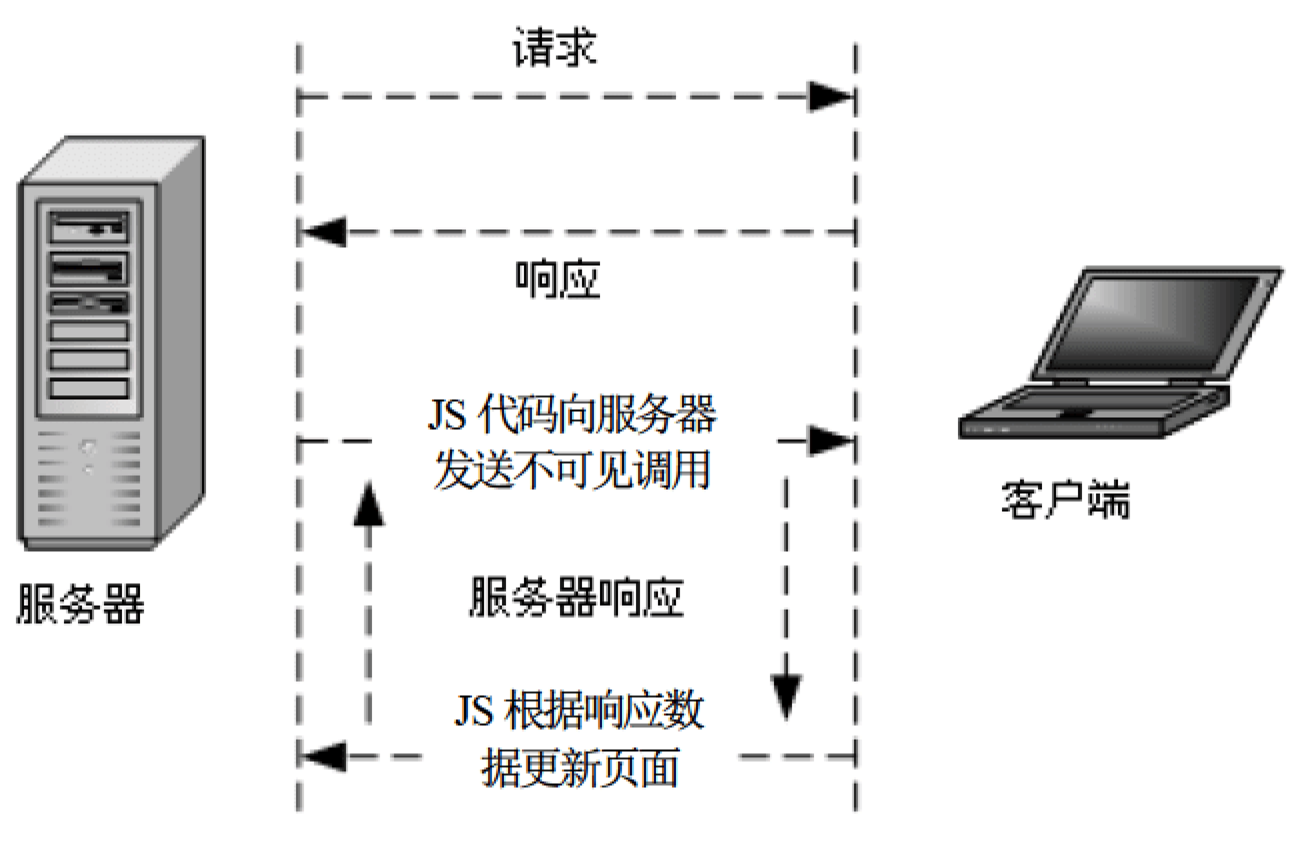


图 23.1　典型的 Ajax 应用

23.1.3 Ajax 的优点

Ajax 在用户与服务器之间引入了 Ajax 引擎作为中间媒介，Web 页面不用打断交互流程就可以重新



加载，实现动态更新，从而可以消除网络交互过程中“处理—等待—处理—等待”的缺点。也就是说，不需刷新客户端浏览器就可以实现重新向服务器发出请求。

使用 Ajax 的优点具体表现在以下几个方面：

☑ 无刷新更新页面。Ajax 使用 XMLHttpRequest 对象向服务器发送请求并得到服务器响应，在不需要重新载入整个页面的情况下，即可根据服务器端返回的请求更改页面内容，使用 Ajax 技术可以有效减少用户等待页面刷新的时间。

☑ 可以把一部分以前由服务器负担的工作转移到客户端，利用客户端闲置的资源进行处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。

☑ 减轻服务器的负担。Ajax 的原则是“按需求获取数据”，可以最大程度地减少冗余请求和响应对服务器造成的负担。

☑ 可以调用 XML 等外部数据，进一步促进 Web 页面显示和数据的分离。

☑ Ajax 是基于标准化并被广泛支持的技术，不需要下载插件或者小程序。

23.2　Ajax 技术的组成

Ajax 技术并不是一种新型的语言，而是多种已有的 Web 技术的整合，主要包括 JavaScript、XML 语言、DOM 和 CSS 等。

23.2.1　JavaScript 脚本语言

JavaScript 是一种解释型的、基于对象的脚本语言，其核心已经嵌入到目前主流的 Web 浏览器中。虽然平时应用最多的是通过 JavaScript 实现一些网页特效及表单数据验证等功能，但 JavaScript 可以实现的功能远不止这些。JavaScript 是一种具有丰富的面向对象特性的程序设计语言，利用它能执行许多复杂的任务，例如，Ajax 就是利用 JavaScript 将 DOM、XHTML（或 HTML）、XML 以及 CSS 等技术综合起来，并控制它们的行为。因此，要开发一个复杂高效的 Ajax 应用程序，就必须对 JavaScript 有深入的了解。

JavaScript 不是 Java 语言的精简版，并且只能在某个解释器或“宿主”上运行，如 ASP、PHP、JSP、Internet 浏览器或者 Windows 脚本宿主。

JavaScript 是一种宽松类型的语言，宽松类型意味着不必显式定义变量的数据类型。此外，在大多数情况下，JavaScript 将根据需要自动进行转换。例如，如果将一个数值添加到由文本组成的某项（一个字符串），该数值将被转换为文本。

23.2.2　XMLHttpRequest 对象

视频讲解：光盘\TM\Video\23\XMLHttpRequest 对象.exe

XMLHttpRequest 是 Ajax 中最核心的技术，它是一个具有应用程序接口的 JavaScript 对象，能够使用超文本传输协议（HTTP）连接一个服务器，是微软公司为了满足开发者的需要，于 1999 年在 IE 5.0 浏览器中率先推出的。现在许多浏览器都对其提供了支持，不过实现方式与 IE 有所不同。使用

XMLHttpRequest 对象，Ajax 可以像桌面应用程序一样只同服务器进行数据层面的交换，而不用每次都刷新页面，也不用每次都将数据处理的工作交给服务器来做，这样既减轻了服务器负担又加快了响应速度，缩短了用户等待的时间。

使用 XMLHttpRequest 对象发送请求和处理响应之前，首先需要初始化该对象，由于 XMLHttpRequest 不是一个 W3C 标准，所以对于不同的浏览器，初始化的方法也是不同的。

☑ IE 浏览器

IE 浏览器把 XMLHttpRequest 实例化为一个 ActiveX 对象。具体方法如下：

```
var http_request = new ActiveXObject("Msxml2.XMLHTTP");
```

或者

```
var http_request = new ActiveXObject("Microsoft.XMLHTTP");
```

上面语法中的 Msxml2.XMLHTTP 和 Microsoft.XMLHTTP 是针对 IE 浏览器的不同版本而进行设置的，目前比较常用的是这两种。

☑ Mozilla、Safari 等其他浏览器

Mozilla、Safari 等其他浏览器把它实例化为一个本地 JavaScript 对象。具体方法如下：

```
var http_request = new XMLHttpRequest();
```

为了提高程序的兼容性，可以创建一个跨浏览器的 XMLHttpRequest 对象。方法很简单，只需要判断一下不同浏览器的实现方式，如果浏览器提供了 XMLHttpRequest 类，则直接创建一个实例，否则使用 IE 的 ActiveX 控件。具体代码如下：

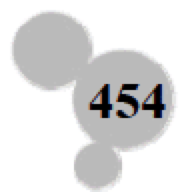
```
if (window.XMLHttpRequest) {                                    //Mozilla、Safari 等浏览器
    http_request = new XMLHttpRequest();
}
else if (window.ActiveXObject) {                                //IE 浏览器
    try {
        http_request = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            http_request = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e) {}
    }
}
```

由于 JavaScript 具有动态类型特性，而且 XMLHttpRequest 对象在不同浏览器上的实例是兼容的，所以可以用同样的方式访问 XMLHttpRequest 实例的属性和方法，不需要考虑创建该实例的方法。下面介绍 XMLHttpRequest 对象的常用方法和属性。

1. XMLHttpRequest 对象的常用方法

（1）open()方法

open()方法用于设置进行异步请求目标的 URL、请求方法以及其他参数信息，具体语法如下：



```
open("method","URL"[,asyncFlag[,"userName"[, "password"]]])
```

在上面的语法中，method 用于指定请求的类型，一般为 get 或 post；URL 用于指定请求地址，可以使用绝对地址或者相对地址，并且可以传递查询字符串；asyncFlag 为可选参数，用于指定请求方式，同步请求为 true，异步请求为 false，默认为 true；userName 为可选参数，用于指定用户名，没有时可省略；password 为可选参数，用于指定请求密码，没有时可省略。

（2）send()方法

send()方法用于向服务器发送请求。如果请求声明为异步，该方法将立即返回，否则将等到接收到响应为止。具体语法格式如下：

```
send(content)
```

在上面的语法中，content 用于指定发送的数据，可以是 DOM 对象的实例、输入流或字符串，如果没有参数需要传递可以设置为 null。

（3）setRequestHeader()方法

setRequestHeader()方法为请求的 HTTP 头设置值。具体语法格式如下：

```
setRequestHeader("label", "value")
```

在上面的语法中，label 用于指定 HTTP 头，value 用于为指定的 HTTP 头设置值。

注意： *setRequestHeader()方法必须在调用 open()方法之后才能调用。*

（4）abort()方法

abort()方法用于停止当前异步请求。

（5）getAllResponseHeaders()方法

getAllResponseHeaders()方法用于以字符串形式返回完整的 HTTP 头信息，当存在参数时，表示以字符串形式返回由该参数指定的 HTTP 头信息。

2．XMLHttpRequest 对象的常用属性

XMLHttpRequest 对象的常用属性如表 23.1 所示。

表 23.1　XMLHttpRequest 对象的常用属性

属　性	说　明
onreadystatechange	每次状态改变都会触发这个事件处理器，通常会调用一个 JavaScript 函数
readyState	请求的状态。有以下 5 个取值： 0=未初始化 1=正在加载 2=已加载 3=交互中 4=完成
responseText	服务器的响应，表示为字符串

续表

属　　性	说　　明
responseXML	服务器的响应，表示为 XML。这个对象可以解析为一个 DOM 对象
status	返回服务器的 HTTP 状态码，如： 200="成功" 202="请求被接受，但尚未成功" 400="错误的请求" 404="文件未找到" 500="内部服务器错误"
statusText	返回 HTTP 状态码对应的文本

3．XMLHttpRequest 对象与服务器交互

XMLHttpRequest 对象最大的用途就是不需要刷新页面就可以与服务器进行交互。可以将 Ajax 与服务器的交互分为以下 3 个步骤：

（1）初始化 XMLHttpRequest 对象，关键代码如下。

```
var xmlHttp = false;                          //定义 XMLHttpRequest 对象
try {
                                              //如果浏览器支持 XMLHttpRequest 对象，创建 ActiveXObject 对象
    xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
    try {
        xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e2) {}
}
if (!xmlHttp && typeof XMLHttpRequest != "undefined") {
    try{
        xmlHttp = new XMLHttpRequest();
    }catch(e3){ xmlHttp = false;}
}
```

（2）设置请求状态和返回处理函数，语法格式如下。

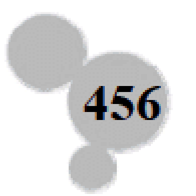
```
xmlobj.onreadystatechange=function_name;
```

其中，xmlobj 是 XMLHttpRequest 的对象；function_name 是用来处理请求状态和返回码的函数名（即回调函数）。

（3）发送 HTTP 请求，语法格式如下。

```
xmlobj.open(send_method,url,flag);
```

其中，xmlobj 是 XMLHttpRequest 的对象；send_method 是发送方法，可以是 GET 或者 POST，对应表单使用的方法；url 是页面要调用的地址；flag 是一个标记，如果为 true 则表示在等待被调用页面响应的时间内可以继续执行页面代码，反之为 false。



例 23.1　通过 XMLHttpRequest 对象读取 HTML 文件，并输出读取结果。关键代码如下：（**实例位置：光盘\TM\Instances\23\1**）

```
<script  langurage="javascript">
var xmlHttp;                                          //定义 XMLHttpRequest 对象
function createXmlHttpRequestObject(){
    if(window.ActiveXObject){
        try{
            xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
        }catch(e){
            xmlHttp=false;
        }
    }else{                                            //如果在 Mozilla 或其他的浏览器下运行
        try{
            xmlHttp=new XMLHttpRequest();
        }catch(e){
            xmlHttp=false;
        }
    }
    if(!xmlHttp)
        alert("返回创建的对象或显示错误信息");
    else
        return xmlHttp;
}
function ReqHtml(){
    createXmlHttpRequestObject();
    xmlHttp.onreadystatechange=StatHandler;           //判断 URL 调用的状态值并处理
    xmlHttp.open("GET","text.html",true);             //调用 text.html
    xmlHttp.send(null);
}
function StatHandler(){
    if(xmlHttp.readyState==4 && xmlHttp.status==200){
        document.getElementById("webpage").innerHTML=xmlHttp.responseText;
    }
}
</script>
<!--创建超链接-->
<a href="#" onclick="ReqHtml();">通过 XMLHttpRequest 对象请求 HTML 文件</a>
<!--通过 div 标签输出请求内容-->
<div id="webpage"></div>
```

本实例通过 DIV 标签输出请求的 HTML 页面。在 JavaScript 脚本中，应用 document.getElementById()方法来获得页面元素。运行本实例，单击“通过 XMLHttpRequest 对象请求 HTML 文件”超链接，将输出如图 23.2 所示的页面。

通过 XMLHttpRequest 对象不但可以读取 HTML 文件，还可以读取文本文件、XML 文件，其实现交互的方法与读取 HTML 文件类似，这里不再举例。

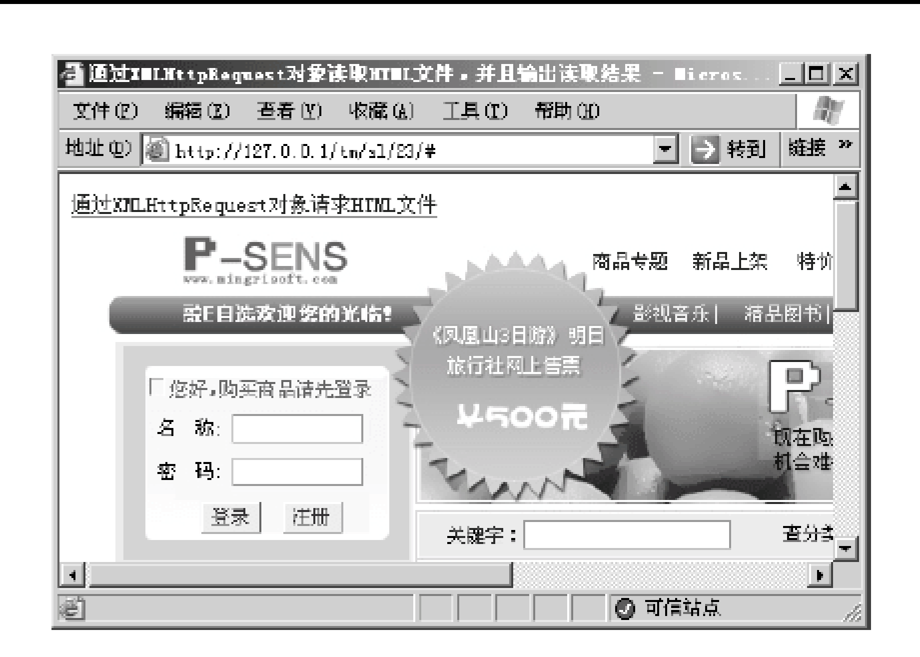


图 23.2　通过 XMLHttpRequest 对象读取 HTML 文件

23.2.3　XML 语言

XML 是 eXtensible Markup Language（可扩展的标记语言）的缩写，它提供了用于描述结构化数据的格式。XMLHttpRequest 对象与服务器交换的数据，通常采用 XML 格式，但也可以是基于文本的其他格式。

23.2.4　DOM

DOM 是 Document Object Model（文档对象模型）的缩写，它为 XML 文档的解析定义了一组接口。解析器读入整个文档，然后构建一个驻留内存的树结构，最后通过 DOM 可以遍历树以获取来自不同位置的数据，可以添加、修改、删除、查询和重新排列树及其分支。另外，还可以根据不同类型的数据源来创建 XML 文档。在 Ajax 应用中，通过 JavaScript 操作 DOM，可以达到在不刷新页面的情况下实时修改用户界面的目的。

23.2.5　CSS

CSS 是 Cascading Style Sheet（层叠样式表）的缩写，是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。在 Ajax 中，通常使用 CSS 进行页面布局，并通过改变文档对象的 CSS 属性控制页面的外观和行为。CSS 是一种 Ajax 开发人员所需要的重要武器，提供了从内容中分离应用样式和设计的机制。虽然 CSS 在 Ajax 应用中扮演至关重要的角色，但它也是构建创建跨浏览器应用的一大阻碍，因为不同的浏览器厂商支持不同的 CSS 级别。

23.3　Ajax 与 PHP 的交互

在 23.2 节介绍 XMLHttpRequest 对象时，讲解了如何实现 Ajax 与 HTML 的交互。这里将介绍 XMLHttpRequest 对象与 PHP 的交互。通过 XMLHttpRequest 对象请求 PHP 页面有两种方式，一种是 GET 方式，另一种是 POST 方式。具体使用哪一种方式与 PHP 进行交互，取决于 PHP 页面中变量值的传递方法，如果使用 POST 方法进行传递变量值，则使用 POST 方式进行交互操作，否则使用 GET



方式进行交互操作。

23.3.1　通过 GET 方式与 PHP 进行交互

视频讲解：光盘\TM\Video\23\通过 GET 方式与 PHP 进行交互.exe

通过一个具体的实例来讲解如何通过 GET 方式与 PHP 进行交互。

例 23.2　应用 GET 方式，通过 XMLHttpRequest 对象与 PHP 进行交互，其操作步骤如下：**（实例位置：光盘\TM\Instances\23\2）**

（1）创建 index.php 文件，编写 JavaScript 脚本，通过 Ajax 请求 searchrst.php 文件，执行查询操作，将查询结果定义到 DIV 标签中；创建 form 表单，提交查询的关键字，通过 DIV 标签输出查询结果，代码如下。

```
<script>
var xmlHttp;                                    //定义 XMLHttpRequest 对象
function createXmlHttpRequestObject(){
    if(window.ActiveXObject){                   //如果在 Internet Explorer 下运行
        try{
            xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
        }catch(e){
            xmlHttp=false;
        }
    }else{                                      //如果在 Mozilla 或其他的浏览器下运行
        try{
            xmlHttp=new XMLHttpRequest();
        }catch(e){
            xmlHttp=false;
        }
    }
    if(!xmlHttp)                                //返回创建的对象或显示错误信息
        alert("返回创建的对象或显示错误信息");
        else
        return xmlHttp;
}
function showsimple(){
    createXmlHttpRequestObject();
    var cont = document.getElementById("searchtxt").value;
    if(cont==""){
        alert('查询关键字不能为空！');
        return false;
    }
    xmlHttp.onreadystatechange=StatHandler;     //判断 URL 调用的状态值并处理
    xmlHttp.open("GET",'searchrst.php?cont='+cont,false);
    xmlHttp.send(null);
}
function StatHandler(){
    if(xmlHttp.readyState==4 && xmlHttp.status==200){
        document.getElementById("webpage").innerHTML=xmlHttp.responseText;
```

```
            $sqles="select * from tb_administrator ";
            $results=mysql_query($sqles,$conn);
            echo  "<table  width='500'  border='1'  cellpadding='1'  cellspacing='1'  bordercolor='#FFFFCC'  bgcolor=
'#666666'>";
            echo "<tr><td height='30' align='center' bgcolor='#FFFFFF'>ID</td><td align='center' bgcolor='#FFFFFF'>名
称</td><td align='center' bgcolor='#FFFFFF'>编号</td><td align='center' bgcolor='#FFFFFF'>描述</td></tr>";
            while($myrow=mysql_fetch_array($results)){                //循环输出查询结果
                  echo "<tr><td height='22' bgcolor='#FFFFFF'>".$myrow[id]."</td>";
                  echo "<td bgcolor='#FFFFFF'>".$myrow[user]."</td>";
                  echo "<td bgcolor='#FFFFFF'>".$myrow[number]."</td>";
                  echo "<td bgcolor='#FFFFFF'>".$myrow[explains]."</td>";
                  echo "</tr>";
            }
            echo "</table>";
      }
?>
```

运行本实例，无刷新添加员工信息后，将在当前页中输出数据表中的所有员工信息，运行结果如图 23.4 所示。



图 23.4　通过 POST 方式与 PHP 进行交互

注意： 本实例中，在获取 Ajax 中 POST 提交的数据时，需要对数据的编码格式进行转换，才能够将中文字符串添加到数据表中。因为 Ajax 中的数据使用的是 UTF-8 格式的编码，如果要将该数据添加到编码格式为 GB2312 的数据表中，就需要使用 iconv()函数将 UTF-8 编码转换为 GB2312 编码。

23.4　Ajax 开发常见问题

视频讲解：光盘\TM\Video\23\Ajax 开发需要注意的几个问题.exe

Ajax 在开发过程中需要注意以下几个问题：

1．浏览器兼容性问题

Ajax 使用了大量的 JavaScript 和 Ajax 引擎，而这些内容需要浏览器提供足够的支持。目前提供这

些支持的浏览器有 IE 5.0 及以上版本、Mozilla 1.0、NetScape 7 及以上版本。Mozilla 虽然也支持 Ajax，但是提供 XMLHttpRequest 对象的方式不一样。所以，使用 Ajax 的程序必须测试其针对各个浏览器的兼容性。

2．XMLHttpRequest 对象封装

Ajax 技术的实现主要依赖于 XMLHttpRequest 对象，但是在调用其进行异步数据传输时，由于 XMLHttpRequest 对象的实例在处理事件完成后就会被销毁，所以如果不对该对象进行封装处理，在下次需要调用它时就得重新构建，而且每次调用都需要写一大段的代码，使用起来很不方便。不过，现在很多开源的 Ajax 框架都提供了对 XMLHttpRequest 对象的封装方案，其详细内容这里不作介绍，读者可参考相关资料。

3．性能问题

由于 Ajax 将大量的计算从服务器移到了客户端，这就意味着浏览器将承受更大的负担，而不再是只负责简单的文档显示。由于 Ajax 的核心语言是 JavaScript，而 JavaScript 并不是以高性能而知名，另外，JavaScript 对象也不是轻量级的，特别是 DOM 元素耗费了大量的内存，因此，如何提高 JavaScript 代码的性能对于 Ajax 开发者来说尤为重要。下面介绍 3 种优化 Ajax 应用执行速度的方法：

☑ 优化 for 循环。
☑ 将 DOM 节点附加到文档上。
☑ 尽量减少点“.”号操作符的使用。

4．中文编码问题

Ajax 不支持多种字符集，它默认的字符集是 UTF-8，所以在应用 Ajax 技术的程序中应及时进行编码转换，否则程序中出现的中文字符将变成乱码。一般来说，以下两种情况会产生中文乱码。

☑ PHP 发送中文、Ajax 接收。只需在 PHP 顶部添加如下语句。

```
header('Content-type: text/html;charset=GB2312');                    //指定发送数据的编码格式
```

XMLHttp 会正确解析其中的中文。

☑ Ajax 发送中文、PHP 接收。这个比较复杂，Ajax 中先用 encodeURIComponent 对要提交的中文进行编码，在 PHP 页中添加如下代码。

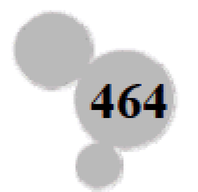
```
$GB2312string=iconv( 'UTF-8', 'gb2312//IGNORE' , $RequestAjaxString);
```

PHP 选择 MySQL 数据库时，应用如下语句设置数据库的编码类型。

```
mysql_query("set names gb2312");
```

23.5 实　　战

Ajax 作为 Web 2.0 时代的关键技术，已经被广泛地应用，并且很多组织已经推出基于 Ajax 的框架，如 jQuery 就是非常快捷、方便的 Ajax 框架，使用 jQuery 技术可以用简单的几行代码完成非常复杂的 Ajax 交互。本节将通过具体实例讲解 Ajax 技术的应用。



是否为空检测按钮，代码如下。

```
<form name="form_register">
    用户名：<input type="text" id="username" name="username" size="20" /> 
    <input type="button" value="查看用户名是否被占用"
    onclick="javascript:chkUsername(form_register.username.value)" />
</form>
```

（3）建立与 MySQL 数据库的连接，选择数据库并设置字符集。

```
    <?php
    $host = '127.0.0.1';                    //MySQL 数据库服务器地址
    $userName = 'root';                     //用户名
    $password = 'root';                     //密码
    $connID = mysql_connect($host, $userName, $password);      //建立与数据库的连接
    mysql_select_db('db_database23', $connID);                 //选择数据库
    mysql_query('set names gbk');           //设置字符集
    ?>
```

（4）建立 chk.php 文件，该文件中的代码用于判断客户端通过 GET 方法提交的用户名的值，并判断该值是否存在，如果存在则返回“y”，否则返回“n”。

```
<?php
require_once 'conn.php';                    //包含数据库连接文件
$sql = mysql_query("select id, username from tb_user where username='".trim($_GET['username'])."'", $connID);
                                            //执行查询
$result = mysql_fetch_array($sql);
if ($result) {                              //判断用户名是否存在
    echo 'y';
} else {
    echo 'n';
}
?>
```

运行上述实例，结果如图 23.5 所示，在用户名文本框中随便输入一个用户名，然后单击“查看用户名是否被占用”按钮，在不刷新页面的情况下即可弹出该用户名是否被占用的提示信息。



图 23.5　检测用户名是否被占用



23.5.2　在 PHP 中应用 Ajax 技术实现博客文章类别添加

视频讲解：光盘\TM\Video\23\在 PHP 中应用 Ajax 技术实现博客文章类别添加.exe

传统方式中，通过 POST 方法提交表单中的数据，页面需要刷新，而使用 Ajax 技术通过 POST 方法提交数据，不需要刷新页面即可实现与服务器的交互，这样可以有效减少刷新页面的等待时间。

例 23.5　使用 Ajax 技术通过 POST 方法添加文章类别。（**实例位置：光盘\TM\Instances\23\5**）

本实例的实现步骤如下：

（1）建立类别信息录入表单。与传统的表单相比，使用 Ajax 技术通过 POST 方法提交数据时，不需要在客户端表单中使用<form>标记，数据类型通过 XMLHttpRequest 对象的 setRequestHeader()方法设定。类别录入表单结构如表 23.2 所示。

表 23.2　类别录入表单的表单元素

元素类型	元素名称	属性设置	说明
文本域	typename	type="text" name="typename" size="30"	博客类别
文本域	username	type="text" name="username" size="30"	发布人
提交按钮	button	type="button" value="保存" onclick="saveType(typename.value, username.value)"	提交按钮

（2）建立 fun.js 脚本文件，在该文件中编写用于将客户端录入的数据通过 POST 方法提交到服务器的 Ajax 代码，如下所示。

```
function saveType(typename, username){
    if(typename==''){                                          //判断博客类别是否为空
        alert('请输入类别名称！');
    }else if(username==''){                                    //判断用户名是否为空
        alert('请输入用户名！');
    }else{
        var xmlObj;                                            //定义 XMLHttpRequest 对象
        var urlData = "typename="+typename+"&username="+username;        //指定要发送的数据
         if(window.ActiveXObject){
//判断浏览器是否支持 ActiveXObject 组件，如果支持则通过 ActiveXObject 方式创建 XMLHttpRequest 对象
             xmlObj = new ActiveXObject("Microsoft.XMLHTTP");
         }else if(window.XMLHttpRequest){
//如果浏览器不支持 ActiveXObject 组件，则使用 XMLHttpRequest 组件创建 XMLHttpRequest 对象
             xmlObj = new XMLHttpRequest();
         }
         xmlObj.onreadystatechange = callBackFun;              //指定回调函数
         xmlObj.open("POST", "saveType.php", true);            //指定提交方法和页面
         xmlObj.setRequestHeader("Content-Type", "application/x-www-form-urlencoded;");
            //指定发送数据库类型
         xmlObj.send(urlData);                                 //指定发送的数据
         function callBackFun(){                               //定义回调函数
             if(xmlObj.readyState == 4 && xmlObj.status == 200){
                 //如果服务器端返回内容并无错误发生
```

```
                if(xmlObj.responseText == 'y'){                //判断保存是否成功，并给出提示
                    alert('类别添加成功！ ');
                }else{
                    alert('类别添加失败！ ');
                }
            }
        }
    }
}
```

上述代码中，首先判断是否已经录入了博客类别信息，并给出相关提示，之后定义并创建XMLHttpRequest对象，使用该对象的onreadystatechange属性指定回调函数，open方法指定通过POST方法提交以及提交页面，setRequestHeader()方法指定发送数据类型，send()方法指定要发送的数据。在回调函数中根据返回参数判断类别是否保存成功，并通过提示窗口给出相应的提示信息。

（3）编写conn.php文件，该文件中的代码用于实现与MySQL数据库的连接，并设置字符集，代码如下。

```
<?php
$host = '127.0.0.1';                                        //MySQL 数据库服务器地址
$userName = 'root';                                         //用户名
$password = 'root';                                         //密码
$connID = mysql_connect($host, $userName, $password);       //建立与数据库的连接
mysql_select_db('db_database23', $connID);                  //选择数据库
mysql_query('set names gbk');                               //设置字符集
?>
```

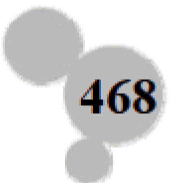
为了防止数据库的中文内容发送乱码，在完成数据选择后通过mysql_query()函数执行SQL命令，设置数据所使用的字符集为GBK编码。

（4）接收通过POST方法提交的内容，并将博客类别等相关内容保存到数据库中。

```
<?php
require_once 'conn.php';                                    //包含数据库连接文件
$typename = iconv('utf-8', 'gbk',$_POST['typename']);       //将类别名称由 UTF-8 编码转变为 GBK 编码
$username = iconv('utf-8', 'gbk', $_POST['username']);      //将用户名称由 UTF-8 编码转变为 GBK 编码
if(mysql_query("insert into tb_type(typename, username, addtime) values('".$typename."', '".$username."', '".date('Y-m-d
H:i:s')."')", $connID)){                                    //将数据库保存到数据库中
    echo 'y';
}else{
    echo 'n';
}
?>
```

上述代码中，使用require_once语句包含conn.php文件从而建立与数据库的连接，由于Ajax在通过POST方法提交数据时默认采用UTF-8编码，所以在接收提交的相关信息后需要使用iconv()函数将所接收的内容由UTF-8编码转换为GBK编码，完成转码后使用mysql_query()函数将类别信息保存到数据库中，如果成功保存则输出y，否则输出n。

运行本实例，结果如图23.6所示，在“请输入博客类别”和“发布人”文本框中填写相关信息后，



```
            $.get("returnpc.php?flag=p", null, function(data){
            //向服务器发送 GET 请求，获取省份的值，并将结果追加到“省份”下拉列表框中
                $("#p").append(data);
            });
            $("#c").css("display","none");              //初始状态使“城市”下拉列表框不可见
            $("#p").change(function(){                  //为“省份”下拉列表框增加改变事件
                if($("#p").val()==""){                  //在未选择省份的情况下，使“城市”下拉列表框不可见
                    $("#c").css("display","none");
                }else{
                    $.get("returnpc.php?flag=c&p="+$("#p").val(), null, function(data){
    //如果选择了某省份，则向服务器发送 GET 请求，使用回调函数为“城市”下拉列表框赋值，并使“城市”下拉列表框可见
                        $("#c").css("display","");
                        $("#c").empty();
                        $("#c").append(data);           //将数据追加到“城市”下拉列表框
                            });
                }
            });
        }
);
```

上述代码中，主要使用 Ajax 的 jQuery 框架，没有 jQuery 框架基础的读者可能不容易理解上述代码，下面对该内容进行介绍。

$(document).ready()表示在页面载入时所执行的操作，是 window.load 注册事件的替代方法。在 ready 方法中使用一个匿名函数作为参数，所有初始操作都在该匿名函数中定义。

$.get()方法用于向服务器发送 GET 请求，该方法的语法格式如下：

```
$.get(url, data, callback);
```

其中，参数 url 为待载入页面的 URL 地址，data 为要向服务器端发送的数据，callback 为载入成功时的回调函数。可以发现，如果直接使用 Ajax 向服务器发送 GET 请求，需要编写多行代码才能实现，而使用 jQuery 框架只需一个方法即可实现，所以 jQuery 确实为程序开发带来了极大的方便。在回调函数中定义如下代码：

```
$("#p").append(data)
```

其中，$("#p")用来获取 HTML 页面中 ID 为 p 的元素的引用，相当于 document.getElementById("p")，append()方法用于向“省份”下拉列表框中添加列表项。

$("#p").change()方法为“省份”下拉列表框的 change 事件，当选择下拉列表框中的选项时触发该事件。在“省份”下拉列表框的 change 事件中向服务器发送 GET 请求，并在回调函数中将服务器返回的值追加到“城市”下拉列表框中。

（4）编写用于返回客户端请求的服务器端代码，根据客户端传递参数 flag 的值判断返回省份信息还是城市信息，之后从省份数组或城市数组中提取要返回的内容并返回。

```
<?php
require_once 'pc.php';                          //包含省份数组和城市数组
$flag = $_GET['flag'];                          //用于区分是返回省份信息还是返回城市信息
```



```
$str="";
if($flag == p){                                           //如果参数 flag 的值为 p，则返回省份信息
      for($i=0; $i<count($p); $i++){                      //通过循环整理所有要追加到“省份”下拉列表框的内容
            $str.="<option value=\"".iconv('gbk','utf-8',$p[$i])."\">".iconv('gbk','utf-8',$p[$i])."</option>";
      }
      echo $str;                                          //输出要返回的参数
}else{
      $index = array_search($_GET['p'], $p);              //提取所选择的省份在省份数组中对应的键值
      for($j=0; $j<count($c[$index]); $j++){              //通过循环整理所有要追加到“城市”下拉列表框的内容
        $str.= "<option value=\"".iconv('gbk','utf-8',$c[$index][$j])."\">".iconv('gbk','utf-8',$c[$index][$j])."</option>";
       }
       echo $str;                                         //输出要返回的参数
}
?>
```

注意：为了防止返回的信息乱码，需要使用 iconv()函数将 GBK 编码转换为 UTF-8 编码。

运行上述实例，结果如图 23.7 所示。默认“城市”下拉列表框不可见，当选择某省份时会显示“城市”下拉列表框，并将该省份所包括的城市显示在“城市”下拉列表框中。



图 23.7　选择地区

23.6　本 章 小 结

本章从 Ajax 技术的基本概念开始，采用循序渐进的方式介绍了 Ajax 技术的开发模式、特性及其组成部分，并重点介绍了 Ajax 技术的核心 XMLHttpRequest 对象，以及编写基本的 Ajax 程序的步骤和思路，最后通过具体实例巩固本章所介绍的内容，尤其在本章最后使用一个简单的实例介绍了基于 Ajax 技术的 jQuery 框架的使用方法。通过本章的学习，读者可以理解 Ajax 技术的概念和基本 Ajax 应用的制作方法，能够在此基础上进一步深入学习 Ajax 技术尤其是 jQuery 框架的使用方法。

23.7　实 战 练 习

1．向服务器上传文件或从服务器下载文件时，经常会等待很长时间，为了显示上传或下载进度，可以使用进度条来显示。要求用 jQuery 框架制作如图 23.8 所示的进度条。**（答案位置：光盘\TM\Instances\23\7）**

2．采用图表的形式显示大量的统计数据非常直观，以往在页面中显示的图表都是静态的，如果将

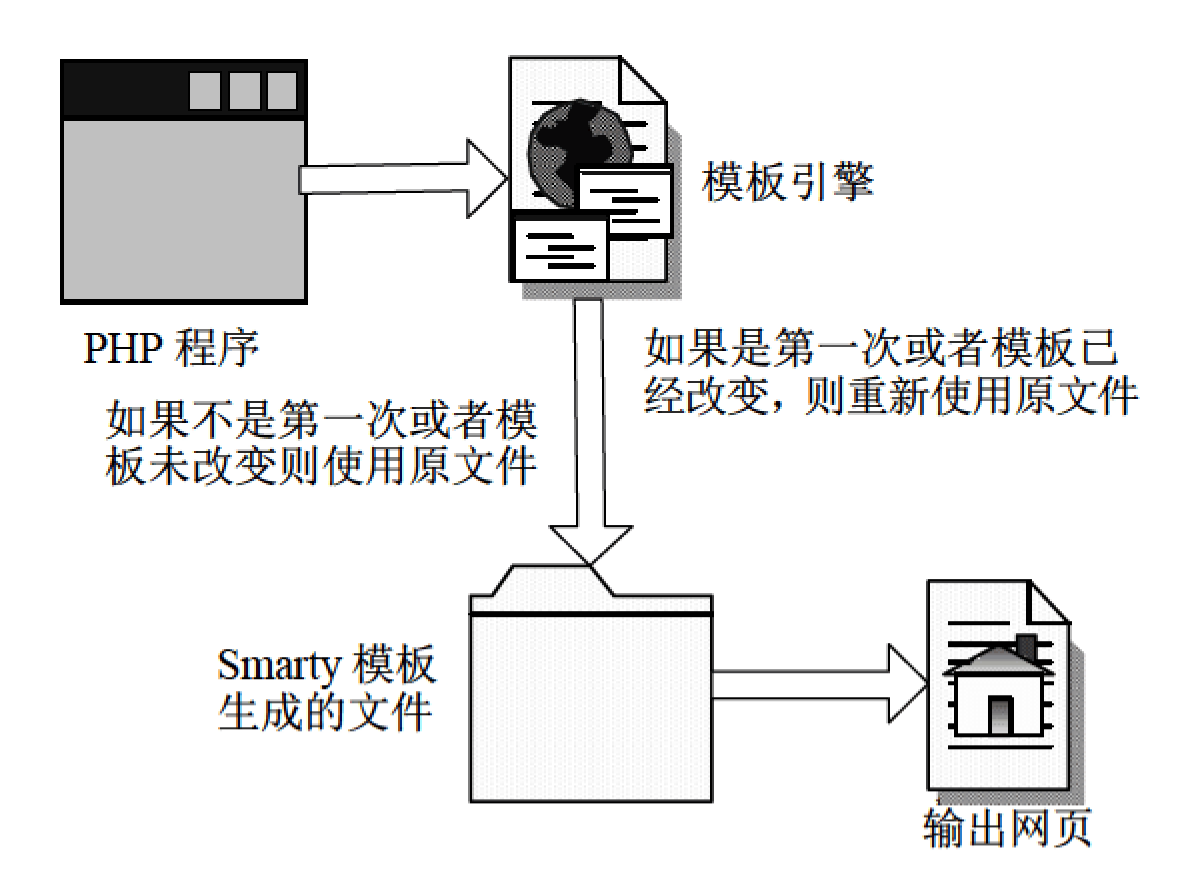


图 24.1　Smarty 模板引擎运行流程图

24.2　Smarty 的安装配置

视频讲解：光盘\TM\Video\24\Smarty 的安装配置.exe

24.2.1　Smarty 下载和安装

PHP 没有内置 Smarty 模板类，需要单独下载和配置，并且 Smarty 要求服务器上的 PHP 版本最低为 4.0.6。用户可以通过 http://smarty.php.,net/download.php 网站下载最新的 Smarty 压缩包。本章使用的版本是 Smarty-2.6.19。

将压缩包解压后，有一个 libs 目录，其中包含了 Smarty 类库的核心文件，包括 smarty.class.php、smarty_Compiler.class.php、config_File.class.php 和 debug.tpl 4 个文件，还有 internals 和 plug-ins 两个目录。将 libs 目录复制到服务器根目录下，并为其重新命名。一般该目录的名称为 smarty、class 等，这里将 libs 文件夹重新命名为 Smarty。至此，Smarty 模板安装完毕。

注意：凡是在后面的章节中提到 Smarty 类包、Smarty 目录等，都是这个重新命名后的 Smarty 目录，即原 libs 目录。

24.2.2　Smarty 配置

出色地完成 Smarty 的配置工作，是应用 Smarty 模板引擎的关键。

（1）确定 Smarty 目录的位置。因为 Smarty 类库是通用的，每一个项目都可能会使用到它。所以将 Smarty 放到根目录下。因为本章的所有程序都放在 tm/sl/24/文件夹下，所以将/24/作为临时的根目录，Smarty 就放到这个目录下。

（2）新建 4 个目录 templates、templates_c、configs 和 cache。因为目录 templates 存放的是项目的模板，所以有人喜欢将 templates 放到 Smarty 目录外。这两种方法没什么区别，只要设置的路径正确即可。

（3）创建配置文件。如果要应用 Smarty 模板，就一定要包含 Smarty 类库和相关信息。将配置信息写到一个文件中，用的时候只要包含配置文件就可以。这里要注意一点，配置文件中要使用绝对路径，因为服务器不会知道文件在第几层目录中被调用。配置文件完成后，保存到根目录下。不要忘记本章中所指的根目录是/24/。配置文件 config.php 的代码如下。

```
<?php
/*   定义服务器的绝对路径   */
define('BASE_PATH','F:\AppServ\www\\');
/*   定义 Smarty 目录的绝对路径   */
define('SMARTY_PATH','TM\sl\24\Smarty\\');
/*   加载 Smarty 类库文件   */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/*   实例化一个 Smarty 对象   */
$smarty = new Smarty;
/*   定义各个目录的路径      */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs/';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';
/*   定义定界符   */
//$smarty->left_delimiter = '<{';
//$smarty->right_delimiter = '}>';
?>
```

上述配置文件的参数说明如下。

☑ BASE_PATH：指定服务器的绝对路径。

说明：指定服务器绝对路径的目的是找到 Smarty 文件夹在服务器中的存储位置。这里有两种方法可以使用：第一种，直接指定绝对路径，如 F:\AppServ\www\\，使用这种方法来指定服务器的绝对路径，一旦服务器的绝对路径发生更改，就必须要修改配置文件，否则程序就会运行出错；第二种，通过全局变量$_SERVER['DOCUMENT_ROOT']来获取服务器的绝对路径，使用该方法不会因为服务器路径的更改而影响到程序的执行。推荐使用第二种方法定义服务器的绝对路径。

☑ SMARTY_PATH：指定 Smarty 目录的绝对路径。
☑ require()：加载 Smarty 类库文件 Smarty.class.php。
☑ $smarty：实例化 Smarty 对象。
☑ $smarty->template_dir：定义模板目录存储位置。
☑ $smarty-> compile_dir：定义编译目录存储位置。
☑ $smarty-> config_dir：定义 configs 文件存储位置。
☑ $smarty-> cache_dir：定义模板缓存目录。
☑ $smarty->left_delimiter：定义 Smarty 使用的开始定界符。
☑ $smarty->right_delimiter：定义 Smarty 使用的结束定界符。

有关定界符的使用，开发者可以指定任意的格式。也可以不指定定界符，使用 Smarty 默认的定界

符“{”和“}”。

到此，Smarty 的配置讲解完毕。

24.2.3　第一个 Smarty 程序

视频讲解：光盘\TM\Video\24\第一个 Smarty 程序.exe

介绍 Smarty 的下载、安装和配置方法后，接下来就创建一个 Smarty 程序，通过具体的操作来了解 Smarty 的应用。

例 24.1　创建第一个 Smarty 应用实例，初步了解 Smarty 的使用过程。**（实例位置：光盘\TM\Instances\24\1）**

（1）在服务器根目录下的/tm/sl/24/文件夹下，新建一个文件夹，命名为 1。

（2）复制 Smarty 到目录 1 下。在 Smarty 目录下新建 4 个目录，分别是 templates、templates_c、configs 和 cache。此时，例 24.1 的目录结构如图 24.2 所示。

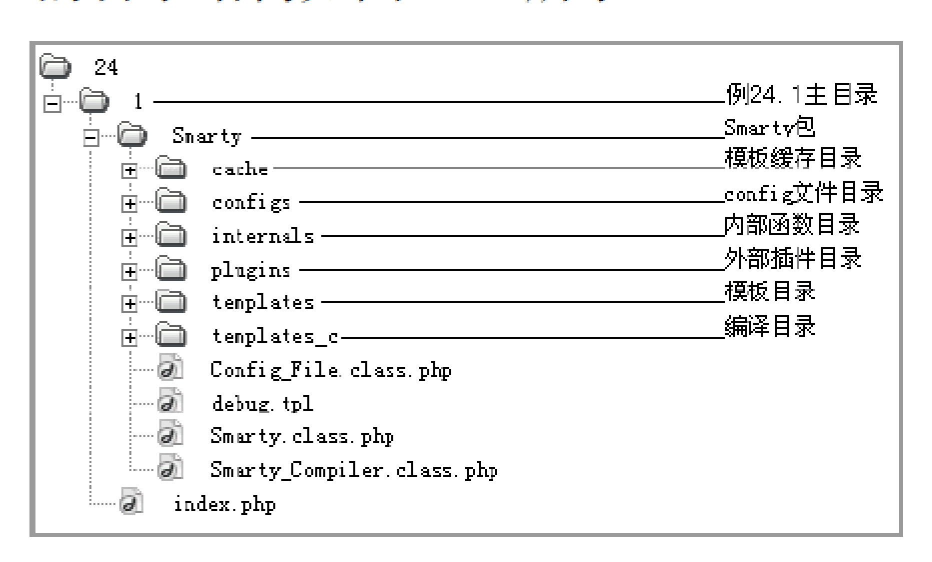


图 24.2　Smarty 包的目录结构

（3）新建一个.html 静态页，输入数据。输入完毕后将文件保存到刚刚新建的 templates 目录下，并命名为 index.tpl。实例代码如下。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{ $title }</title>
</head>
<body>
	{$content}
</body>
</html>
```

说明：代码中加粗的部分就是 Smarty 标签，大括号“{}”为标签的定界符，$title 和$content 为变量。

（4）回到上级目录，在目录/1/下新建一个.php 文件，使用 Smarty 变量和方法对文件进行操作，输入完毕后保存为 index.php，代码如下。

```
<?php
/*  定义服务器的绝对路径  */
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);
```

类库也自然被分成两部分来使用，即 Smarty 模板设计和 Smarty 程序设计。两部分内容既相互独立，同时也有一部分共通。本节首先来学习 Smarty 模板设计。

24.3.1　Smarty 模板文件

Smarty 模板文件是由一个页面中所有的静态元素，加上定界符“{…}”组成的。模板文件统一存放的位置是 templates 目录下（模板文件的存储位置可以在配置文件中指定，可以根据个人习惯而定）。模板中不允许出现 PHP 代码段。Smarty 模板中的所有注释、变量、函数等都要包含在定界符内。

24.3.2　注释

Smarty 中的注释和 PHP 注释类似，都不会显示在源代码中。注释包含在两个星号“*”中间，格式如下：

```
{* 这是注释 *}
```

24.3.3　变量

Smarty 中的变量来自以下 3 个部分：

☑　来自 PHP 页面中的变量，也就是 assign()方法传过来的变量。

使用方法和在 PHP 中是相同的，也需要使用“$”符号。略有不同的是对数组的读取。Smarty 中读取数组有两种方法。一种是通过索引获取，与 PHP 中相似，可以是一维，也可以是多维；另一种是通过键值获取数组元素，这种方法的格式与以前接触过的不同，使用符号“.”作为连接符，数组$arr = array{'object' => 'book','type' => 'comptuer','unit' => '本'}，如果想得到 type 的值，表达式的格式为$arr.type。这个格式同样适用于二维数组。

例 24.2　使用上述两种方法读取数组值，代码如下：**（实例位置：光盘\TM\Instances\24\2）**

```
Smarty/templates/2/index.tpl 文件
<html>
<head>
{*  页面的标题变量$title  *}
<title>{ $title }</title>
</head>
<body>
购书信息：<p>
{*  使用索引取得数组的第一个元素值  *}
图书类别：{ $arr[0] }<br />
{*  使用键值取得第二个数组元素值  *}
图书名称：{ $arr.name }<br />
{*  使用键值取得二维数组的元素值  *}
图书单价：{ $arr.unit_price.price }/{ $arr.unit_price.unit }
</body>
```

运行结果如图 24.5 所示。

☑ 从配置文件中读取数据。Smarty 模板也可以通过配置文件来赋值。对于 PHP 开发人员来说，对配置文件的使用从安装服务器就开始了，对文件的格式也有了一个初步的了解。调用配置文件中变量的格式有以下两种：

- 使用“#”号。将变量名置于两个“#”号中间，即可像普通变量一样调用配置文件内容。
- 使用保留变量中的$smarty_config 来调用配置文件。

例 24.4 通过上面两种方法来调用配置文件 4.conf 的内容，代码如下：**（实例位置：光盘\TM\Instances\24\4）**

```
configs/4/4.conf 文件
title = "调用配置文件"
bgcolor = "#f0f0f0"
border = "5"
type = "计算机类"
name = "PHP 从入门到精通"
templates/04/infex.tpl 文件
{ config_load file="4/4.conf" }
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{#title#}</title>
</head>
<body bgcolor="{#bgcolor#}">
<table border="{#border#}">
<tr>
	<td>{$smarty.config.type}</td>
	<td>{$smarty.config.name}</td>
</tr>
</table>
</body>
</html>
index.php 文件
<?php
	include_once '../config.php';
	$smarty->display('4/index.tpl');
?>
```

运行结果如图 24.6 所示。



图 24.5　Smarty 保留变量

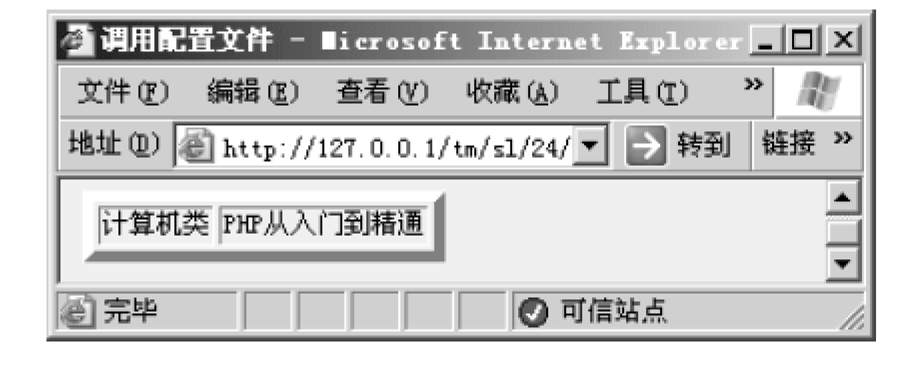


图 24.6　调用配置文件



24.3.4 修饰变量

24.3.3 节中介绍了如何在 Smarty 模板中调用变量。但有的时候，不仅要取得变量值，还要对变量进行处理。变量修饰的一般格式如下：

```
{variable_name|modifer_name: parameter1:…}
```

- ☑ variable_name 为变量名称。
- ☑ modifer_name 为修饰变量的方法名。变量和方法之间使用符号“|”分隔。
- ☑ parameter1 是参数值。如果有多个参数，则使用“:”分开。

Smarty 提供了修饰变量的方法。常用的方法如表 24.2 所示。

表 24.2 修饰变量的常用方法和说明

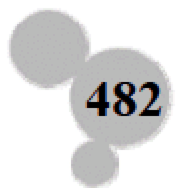
方 法 名	说 明
capitalize	首字母大写
count_characters:true/false	变量中的字符串个数。如果后面有参数 true，则空格也被计算，否则忽略空格
cat: "characters"	将 cat 中的字符串添加到指定字符串的后面
date_format: "%Y-%M-%D"	格式化日期和时间。等同于 PHP 中的 strftime()函数
default: "characters"	设置默认值。当变量为空时，将使用 default 后面的默认值
escape: "value"	用于字符串转码。value 值可以为 html、htmlall、url、quotes、hex、hexentity 和 javascript，默认为 html
lower	将变量字符串改为小写
nl2br	所有的换行符将被替换成 ，功能同 PHP 中的 nl2br()函数一样
regex_replace:"parameter1":"value2"	正则替换。用 value2 替换所有符合 parameter1 标准的字串
replace: "value1":"value2"	替换。使用 value2 替换掉所有 value1
string_format: "value"	使用 value 来格式化字符串。如果 value 为%d，则字符串被格式化为十进制数
strip_tags	去掉所有的 html 标签
upper	将变量改为大写

在对变量进行修饰时，不仅可以单独使用上面的方法，还可以同时使用多个。在每个方法之间使用“|”分隔即可。

例 24.5 使用表 24.2 中的几个方法来修饰字符串。其他方法的使用，读者可以自行练习。实例代码如下：**（实例位置：光盘\TM\Instances\24\5）**

Templates/5/index.tpl 文件

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
<link rel="stylesheet" href="../css/style.css" />
</head>
```



```
<body>
原文：{$str}
<p>
变量中的字符数（包括空格）：{$str|count_characters:true}
<br />
使用变量修饰方法后：{$str|nl2br|upper}
</body>
</html>
index.php 文件
<?php
	include_once "../config.php";
	$str1 = '这是一个实例。';
	$str2 = "\n 图书->计算机类->php\n 书名：《php 从入门到精通》";
	$str3 = "\n 价格：￥59/本。";
	$smarty->assign('title','使用变量修饰方法');
	$smarty->assign('str',$str1.$str2.$str3.$str4);
	$smarty->display('5/index.tpl');
?>
```

运行结果如图 24.7 所示。

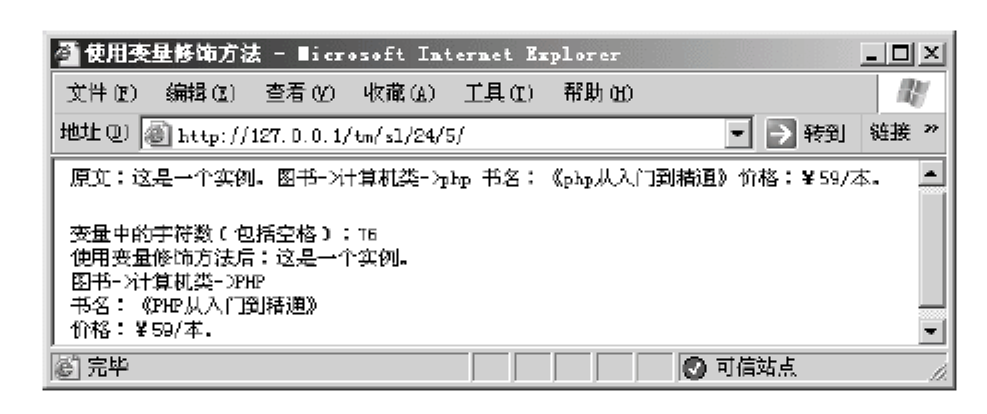


图 24.7　使用变量修饰方法

24.3.5　流程控制

Smarty 模板中的流程控制语句包括 if…elseif…else 条件控制语句和 foreach、section 循环控制语句。

1．if…elseif…else 语句

if 条件控制语句的使用和 PHP 中的 if 大同小异。需要注意的是，if 必须以/if 为结束标记。if 语句的格式如下：

```
{if 条件语句 1}
	语句 1
{elseif 条件语句 2}
	语句 2
{else}
	语句 3
{/if}
```

上述条件语句中，除了可以使用 PHP 中的<、>、=、!=等常见运算符外，还可以使用 eq、ne、neq、gt、lt、lte、le、gte、ge、is even、is odd、is not even、is not odd、not、mod、div by、even by、odd by 等修饰词修饰。具体含义留给读者自己动手来理解。

例 24.6 使用条件判断语句选择不同的返回信息，代码如下：（**实例位置：光盘\TM\Instances\24\6**）

templates/6/index.tpl 文件

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
<link rel='stylesheet' href="../css/style.css" />
</head>
<body>
<p>
{if $smarty.get.type == 'tm'}
欢迎光临，{$smarty.get.type}
{else}
对不起，您不是本站 VIP，无权访问此栏目。
{/if}
</body>
</html>
```

index.php 文件

```
<?php
    include_once "../config.php";
    $smarty->assign("title","if 条件判断语句");
    $smarty->display("6/index.tpl");
?>
```

运行结果如图 24.8 所示。



图 24.8 if 条件判断语句

2．foreach 循环控制

Smarty 模板中的 foreach 语句可以循环输出数组，与另一个循环控制语句 section 相比，在使用格式上要简单得多，一般用于简单数组的处理。foreach 语句的格式如下：

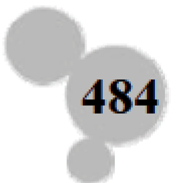
```
{foreach name=foreach_name key=key item=item from=arr_name}
…
{/foreach}
```

其中，name 为该循环的名称；key 为当前元素的键值；item 为当前元素的变量名；from 为该循环的数组。其中，item 和 from 是必要参数，不可省略。

例 24.7 使用 foreach 语句，循环输出数组 infobook 的全部内容，代码如下：（**实例位置：光盘\TM\Instances\24\7**）

templates/7/index.tpl 文件

```
<html>
<head>
```



```
<table width="100" border="0" align="left" cellpadding="0" cellspacing="0">
{section name=sec1 loop=$obj}
    <tr>
        <td colspan="2">{$obj[sec1].bigclass}</td>
    </tr>
    {section name=sec2 loop=$obj[sec1].smallclass}
    <tr>
        <td width="25"> </td>
        <td width="75">{$obj[sec1].smallclass[sec2].s_type}</td>
    </tr>
    {/section}
{/section}
</table>
</body>
</html>
```

index.php 文件

```
<?php
    require "../config.php";
    $obj = array(array("id" => 1, "bigclass" => "计算机图书","smallclass" => array(array("s_id" => 1, "s_type" => "PHP"))),array("id" => 2, "bigclass" => "历史传记","smallclass" => array(array("s_id" => 2, "s_type" => "中国历史"), array("s_id" => 3, "s_type" => "世界历史"))),array("id" => 3, "bigclass" => "畅销小说","smallclass" => array(array("s_id" => 4, "s_type" => "网络小说"),array("s_id" => 5, "s_type" => "科幻小说"))));
     $smarty->assign('title','section 循环控制');
    $smarty->assign("obj", $obj);
    $smarty->display("8/index.tpl");
?>
```

运行结果如图 24.10 所示。

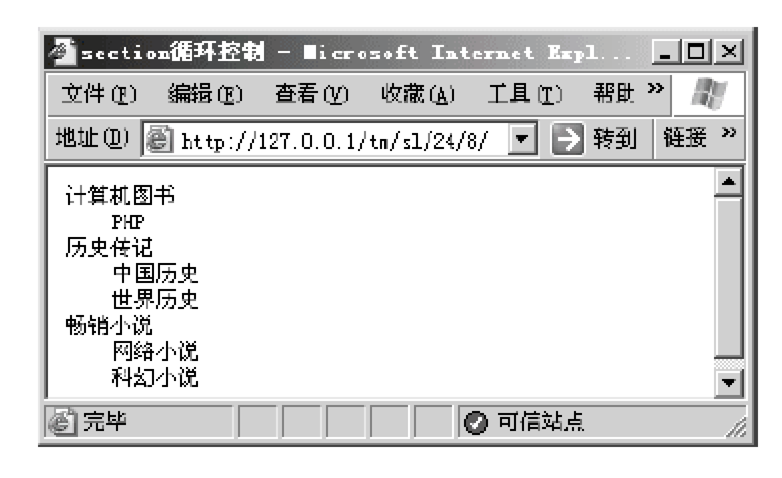


图 24.10　使用 section 循环控制输出数组

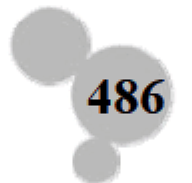
24.3.6　内建函数

Smarty 自带一些内建函数，内建函数是模板语言的一部分，用户不能创建名称与内建函数相同的自定义函数，也不能修改内建函数。下面介绍几个比较常用的内建函数。

1．config_load 函数

从配置文件中加载变量。配置文件有可能包含多个部分，此时可以使用附加属性 section 指定从哪一部分中取得变量：

```
{config_load file="file_name " section="add_attribute"}
```



其中，file_name 指定包含的配置文件的名称；add_attribute 为附加属性，当配置文件中包含多个部分时应用，指定具体从哪一部分中取得变量。

有关该函数的应用可以参考例 24.4。

2．include 标签

include 标签用于在当前模板中包含其他模板，当前模板中的变量在被包含的模板中可用。必须指定 file 属性，该属性指明模板资源的位置。标签语法如下：

```
{include file="file_name " assign=" " var="    "}
```

其中，file 指定包含模板文件的名称；assign 指定一个变量保存包含模板的输出；var 传递给待包含模板的本地参数，只在待包含模板中有效。

3．php 标签

php 标签允许在模板中直接嵌入 PHP 脚本。是否处理这些语句取决于$php_handling 的设置。该语句通常不需要使用，当然如果非常了解此特性或认为有必要，也可以使用。

php 标签的应用演示如下：

```
{php}
        // including a php script directly
        // from the template.
        include("/conn.php");
{/php}
```

上述代码为直接在模板中应用 include 语句调用包含文件。

例 24.9　应用 include 标签在模板页中包含其他模板，并通过 php 标签在模板页中包含其他 PHP 脚本文件。实例代码如下：**（实例位置：光盘\TM\Instances\24\9）**

templates/9/index.tpl 文件

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>{$title}</title>
<link rel="stylesheet" href="../css/style.css" />
</head>
<body>
<!--应用 php 标签通过 include 语句调用包含文件调用-->
{php}
        include("top.php");
{/php}
<table width="99" border="0" cellpadding="0" cellspacing="0">
        <tr>
                <td height="20"><span class="STYLE1">{$arr1.object}</span></td>
        </tr>
        <tr><td height="27">市场价：{$arr1.price}￥</td></tr>
        <tr><td height="22">热卖价：{$arr1.t_price}￥</td></tr>
</table>
<!--通过 include 标签包含 tpl 文件-->
```

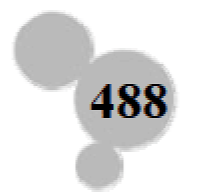
```
{include file="9/bottom.tpl"}
index.php 文件
<?php
    include '../config.php';
    $arr1 = array("object"=>'液晶显示器',"price"=>'3500',"t_price"=>'2888');
    $arr2 = array("object"=>'羽绒服',"price"=>'150',"t_price"=>'88');
    $arr3 = array("object"=>'电视',"price"=>'1000',"t_price"=>'888');
    $smarty->assign('title','应用内建函数');
    $smarty->assign("arr1",$arr1);
    $smarty->assign('arr2',$arr2);
    $smarty->assign('arr3',$arr3);
    $smarty->display('9/index.tpl');
?>
```

在本实例中，通过 php 标签在模板页中直接使用 PHP 的 include 包含语句调用头文件，通过 Smarty 的 include 标签包含 tpl 文件输出页面的尾文件，运行结果如图 24.11 所示。



图 24.11　php 标签的应用

24.4　Smarty 程序设计



视频讲解：光盘\TM\Video\24\Smarty 程序设计.exe

通过前面的学习已经知道，在 Smarty 模板中，是不推荐有 PHP 代码段的，所有的 PHP 程序都要另写成文件。Smarty 程序的功能主要分两种：一种功能是和 Smarty 模板之间的交互，如方法 assign、display；另一种功能就是配置 Smarty，如变量 template_dir、$config_dir 等。本节将学习 Smarty 程序设计的其他一些方法和配置参数。

24.4.1　Smarty 中的常用方法

1．assign

assign 用于在模板被执行时为模板变量赋值。语法如下：

"%%…%%index.tpl.php"格式的文件。随便打开一个这样的文件可以发现，实际上 Smarty 将模板和 PHP 程序又重新组合成一个混编页面。

- ☑ $cache_dir：缓存目录。用来存放缓存文件。同样，在 cache 目录下可以看到生成的.html 文件。如果 caching 变量开启，那么 Smarty 将直接从这里读取文件。
- ☑ $config_dir：配置目录。该目录用来存放配置文件。例 24.4 中所用到的配置文件，就保存到这里。
- ☑ $debugging：调试变量。该变量可以打开调试控制台。只要在配置文件（config.php）中将 $smarty->debugging 设为 true 即可使用。
- ☑ $caching：缓存变量。该变量可以开启缓存。只要当前模板文件和配置文件未被改动，Smarty 就直接从缓存目录中读取缓存文件而不重新编译模板。

24.5 实　　战

24.5.1 通过 Smarty 实现网页的动静分离

视频讲解：光盘\TM\Video\24\通过 Smarty 实现网页的动静分离.exe

Smarty 的功能是实现网页的动静分离，本章中已经对安装、配置和使用 Smarty 的方法进行了详细讲解，下面将应用 Smarty 模板技术真正实现一次网页的动静分离。

例 24.10 应用 Smarty 模板技术，将从数据库中读取的数据输出到网页中。操作步骤如下：**（实例位置：光盘\TM\Instances\24\10）**

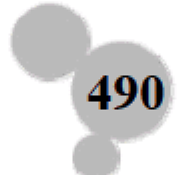
（1）安装和配置 Smarty。将 Smarty 模板的压缩包解压，复制 libs 目录到服务器 tm\sl\24\10\system 目录下，并重新命名为 smarty，在 smarty 目录下新建 4 个目录 templates、templates_c、configs 和 cache，用于存储不同的文件。

编写配置文件，分别指定 4 个目录的存储位置。配置文件 system.smarty.inc.php 的代码如下。

```
<?php
require("smarty/smarty/libs/Smarty.class.php");
class SmartyProject extends   Smarty{
        function SmartyProject(){
                $this->template_dir = "./";                    //指定模板文件存储在根目录下
                $this->compile_dir = "./system/smarty/templates_c/";
                $this->config_dir = "./system/smarty/configs/";
                $this->cache_dir = "./system/smarty/cache/";
        }
}
?>
```

（2）创建数据库操作类。本实例中通过 ADODB 来操作数据库，首先将 ADODB 包放置在 tm\sl\24\10\system 目录下。然后创建数据库连接类，将其存储在 system.class.inc.php 文件中。数据库连接类的代码如下。

```
class ConnDB{                    //数据库连接类
        var $dbtype;
```



```
        var $host;
        var $user;
        var $pwd;
        var $dbname;
        var $debug;
        var $conn;
       function ConnDB($dbtype,$host,$user,$pwd,$dbname,$debug=false){                    //构造方法
                $this->dbtype=$dbtype;
                $this->host=$host;
                $this->user=$user;
                $this->pwd=$pwd;
                $this->dbname=$dbname;
                $this->debug=$debug;
        }
        function GetConnId(){                        //实现数据库的连接并返回连接对象
                require("adodb/adodb.inc.php");
                if($this->dbtype=="mysql" || $this->dbtype=="mssql"){
                        if($this->dbtype=="mysql")
                                $this->conn=NewADOConnection("mysql");
                        else
                                $this->conn=NewADOConnection("mssql");
                                $this->conn->Connect($this->host,$this->user,$this->pwd,$this->dbname);
                }elseif($this->dbtype=="access"){
                                $this->conn=NewADOConnection("access");
                                $this->conn->Connect("Driver={Microsoft Access Driver (*.mdb)};Dbq=".$this->dbname."; Uid=
".$this->user.";Pwd=".$this->pwd.";");
                        }
                $this->conn->Execute("set names gb2312");
                if($this->dbtype=="mysql")
                        $this->conn->debug=$this->debug;
                        return $this->conn;
                }
        function CloseConnId(){                        //关闭与数据库的连接
                $this->conn->Disconnect();
       }
}
```

（3）创建 system.inc.php 文件，实例化数据库连接类，代码如下。

```
<?php
        require("system.smarty.inc.php");
        require("system.class.inc.php");
        //数据库连接类实例化
        $connobj=new ConnDB("mysql","localhost","root","root","db_shop",false);
        $conn=$connobj->GetConnId();
        //调用 Smarty 模板
        $smarty=new SmartyProject();
?>
```

（4）创建 index.php 页面，定义主页面的布局，通过 include 语句调用指定的包含文件，关键代码

如下。

```
<tr>
    <td width="286" height="514" rowspan="2" align="left" valign="top"> </td>
    <td width="648" align="center" valign="top">
        <?php    include('nominate.php');    ?>
     </td>
    <td width="70" rowspan="2" align="center" valign="top"> </td>
</tr>
<tr>
     <td align="center" valign="top">
        <?php    include('newhot.php'); ?>
     </td>
</tr>
```

（5）创建 PHP 脚本文件 nominate.php 和 newhot.php，从数据库中读取数据，并将数据存储到模板变量中，最后指定模板页面。这里只给出 nominate.php 文件中的代码。

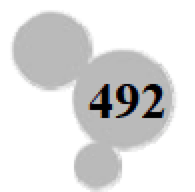
```
<?php
     require("system/system.inc.php");         //调用指定的文件
     $sql = "select id,name,pics,m_price,v_price from tb_commo where isnom = 1 order by id desc";    //定义查询语句
     $num = 4;                                  //定义输出记录数量
     $rst = $conn->SelectLimit($sql,$num);  //执行查询语句
     $nomarr = $rst->GetArray();               //返回查询结果
     $smarty->assign('nomarr',$nomarr);        //将查询结果赋给一个数组
     $smarty->display('nominate.tpl');         //指定模板页面
?>
```

（6）创建 tpl 静态页面 nominate.tpl 和 newhot.tpl，通过 foreach 循环读取出模板变量中的数据，由于篇幅所限，这里没有给出 tpl 文件的代码，可参考光盘中的内容。

运行结果如图 24.12 所示。



图 24.12　通过 Smarty 输出数据库中的数据



24.5.2　Smarty 分页类

视频讲解：光盘\TM\Video\24\Smarty 分页类.exe

上述实例讲解了如何通过 Smarty 实现网页的动静分离，在此基础上讲解如何在 Smarty 中实现分页功能。

例 24.11　应用 Smarty 模板技术，通过分页类将从数据库中读取的数据分页输出。操作步骤如下：**（实例位置：光盘\TM\Instances\24\11）**

（1）首先进行 Smarty 的安装和配置。这里不再赘述，具体操作方法可参考 24.2 节中的内容。

（2）创建数据库连接和管理类。这里不但编写了一个数据库连接类，而且又编写了一个数据库的管理类。通过数据库管理类可以实现对数据库进行查询、修改、删除和更新的操作，代码如下。

```
class AdminDB{                                    //数据库管理类
    function ExecSQL($sqlstr,$conn){              //执行 SQL 语句
        $sqltype=strtolower(substr(trim($sqlstr),0,6));       //截取 SQL 语句中的前 6 个字符串
        $rs=$conn->Execute($sqlstr);
        if($sqltype=="select"){                   //判断如果执行 select 查询
            $array=$rs->GetRows();                //返回查询结果
            if(count($array)==0 || $rs==false)    //如果不执行查询
                return false;                     //返回 false
            else
                return $array;                    //否则返回一个数组
        }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){    //如果执行操作语句
                if($rs)
                    return true;
                else
                    return false;
        }
    }
}
```

数据库连接和管理类的完整代码存储在 system.class.inc.php 文件中。

（3）创建分页类，同样将其存储在 system.class.inc.php 文件中，分页类的代码如下。

```
class SepPage{
    var $rs;
    var $pagesize;
    var $nowpage;
    var $array;
    var $conn;
    var $sqlstr;
    function ShowDate($sqlstr,$conn,$pagesize,$nowpage){ //构造方法
        if(!isset($nowpage) || $nowpage=="")              //判断变量值是否为空
            $this->nowpage=10;                            //定义每页输出的记录数
        else
            $this->nowpage=$nowpage;
```

```
            $this->pagesize=$pagesize;                          //定义每页输出的记录数
            $this->conn=$conn;                                  //连接数据库返回的标识
            $this->sqlstr=$sqlstr;                              //执行的查询语句
            $this->rs=$this->conn->PageExecute($this->sqlstr,$this->pagesize,$this->nowpage);
            @$this->array=$this->rs->GetRows();                 //获取记录数
                if(count($this->array)==0 || $this->rs==false)
                    return false;
                else
                    return $this->array;
        }
        function ShowPage($contentname,$utits,$anothersearchstr,$class){
            $allrs=$this->conn->Execute($this->sqlstr);         //执行查询语句
            $record=count($allrs->GetRows());                   //统计记录总数
            $pagecount=ceil($record/$this->pagesize);           //计算总页数
            $str.="共有".$contentname." ".$record." ".$utits." 每页显示 ".$this->pagesize. " ".
$utits." 第 ".$this->rs->AbsolutePage()." 页/共 ".$pagecount." 页";
            $str.="    ";
            if(!$this->rs->AtFirstPage())
                $str.="<a href=".$_SERVER['PHP_SELF']."?page=1".$anothersearchstr." class=".$class.">首页</a>";
            else
                $str.="<font color='#555555'>首页</font>";
            $str.=" ";
            if(!$this->rs->AtFirstPage())
                $str.="<a
href=".$_SERVER['PHP_SELF']."?page=".($this->rs->AbsolutePage()-1).$anothersearchstr." class=".$class.">上一页</a>";
            else
                $str.="<font color='#555555'>上一页</font>";
            $str.=" ";
            if(!$this->rs->AtLastPage())
                $str.="<a
href=".$_SERVER['PHP_SELF']."?page=".($this->rs->AbsolutePage()+1).$anothersearchstr." class=".$class.">下一页</a>";
            else
                $str.="<font color='#555555'>下一页</font>";
            $str.=" ";
            if(!$this->rs->AtLastPage())
                $str.="<a href=".$_SERVER['PHP_SELF']."?page=".$pagecount.$anothersearchstr." class=".$class.">尾
页</a>";
            else
                $str.="<font color='#555555'>尾页</font>";
            if(count($this->array)==0 || $this->rs==false)
                return "";
            else
                return $str;
        }
}
```

（4）创建 system.inc.php 文件，对数据库连接、操作类以及分页类进行实例化，代码如下。



```
        $arr = $rst->GetAssoc();                                        //获取结果集
        $smarty->assign('arr',$arr);                                    //将返回的结果集存储到指定的 Smarty 模板变量中
        $smarty->display('public.tpl');                                 //执行模板文件
?>
```

（3）在 smarty\templates 文件夹下创建 public.tpl 模板文件，输出模板变量中的公告标题信息，关键代码如下。

```
  <td width="193" align="left" valign="top" class="exam">
            {foreach key=key item=item from=$arr}
            <a  href="#"  class="lk"  onclick="return  showme({$key},'showpub.php');"  ><img  src="images/man.JPG"
width="14" height="11" border="0" />{$item}</a><br />
            {/foreach}
      </td>
```

（4）创建 links.js 文件，定义 showme()函数，通过 JavaScript 脚本调用 showpub.php 文件，查看公告的详细内容。showme()函数的语法如下：

```
// JavaScript Document
function showme(key,wurl){
        var purl = wurl + "?id="+key;
        open(purl,'_blank','width=450 height=200',false);
        return false;
}
```

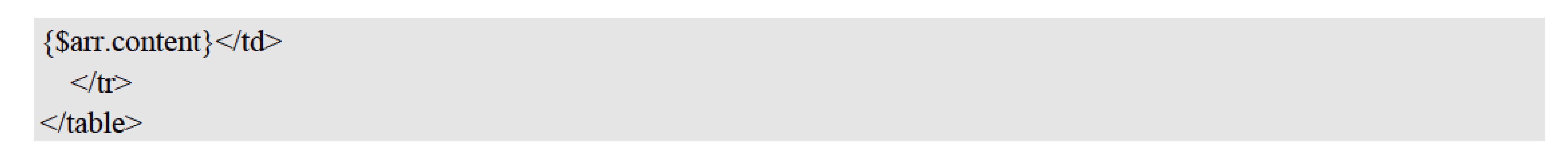
该函数的参数是指定公告信息的 ID 和调用的文件。

（5）创建 showpub.php 文件，根据传递的 ID 值，从数据库中查询出数据的详细信息，并将数据添加到指定的模板变量中，最后指定模板页。

```
<?php
        include_once 'conn/conn.php';
        include_once 'config.php';
        $id = $_GET['id'];
        $sql = "select * from tb_public where id = ".$_GET['id'];
        $rst = $conn->execute($sql);
        $arr = $rst->getAssoc();
        $smarty->assign('title','查看公告');
        $smarty->assign('arr',$arr[$id]);
        $smarty->display('showpub.tpl');
?>
```

（6）创建模板页 showpub.tpl，读取模板变量中的数据，输出公告的详细内容，代码如下。

```
<table width="400" align="center" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="70%" height="25" align="center" valign="middle" class="left">标题：{$arr.title}</td>
    <td width="30%" height="25" align="center" valign="middle" class="right"> {$arr.addtime}</td>
  </tr>
  <tr>
    <td  height="100"  colspan="2"  align="left"  valign="top"  class="all"  style="  text-indent:  10px;"><br> 
```



```
{$arr.content}</td>
  </tr>
</table>
```

运行本实例，将直接输出网站公告的标题信息，单击某个具体的标题，将弹出一个新的窗口，输出具体公告的内容，如图 24.14 所示。

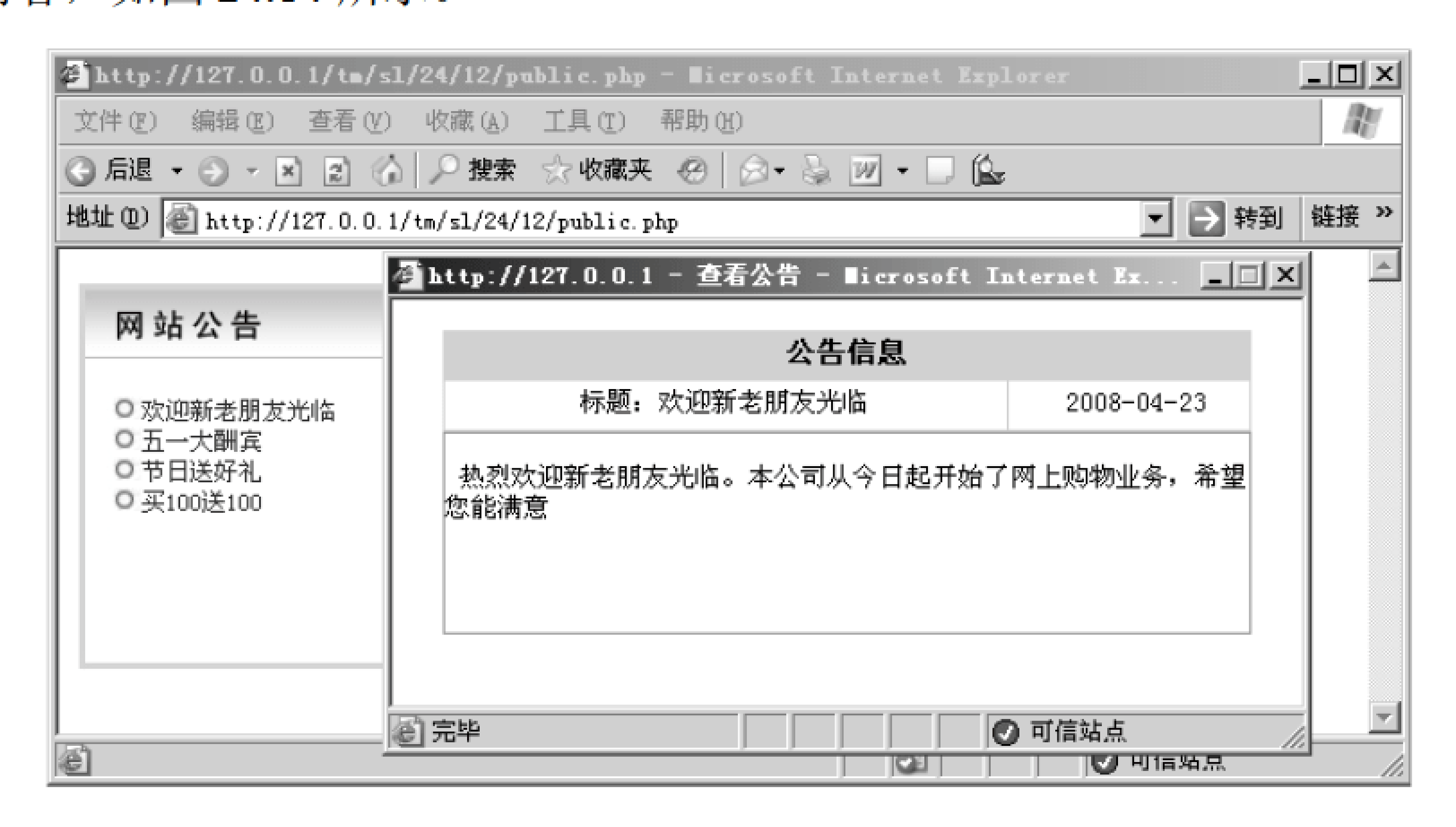


图 24.14　输出公告信息

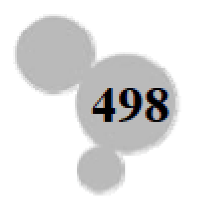
24.6　本 章 小 结

本章主要介绍了 Smarty 模板的安装、配置及使用，并且对 Smarty 的模板设计和程序设计进行了讲解，它们是两个相辅相成的内容，是应用 Smarty 模板的基础。在实战中讲解了 Smarty 的实际应用。希望通过本章的学习，读者能够掌握 Smarty 模板技术，并且能够将其灵活地运用到实际的网站开发中。

24.7　实 战 练 习

1．自定配置一个 Smarty 环境，并将 Smarty 定界符换成其他符号。**（答案位置：光盘\TM\Instances\24\13）**

2．使用 Smarty 模板创建一个购物车。**（答案位置：光盘\TM\Instances\24\14）**



第4部分

项目实战

25.1 项 目 描 述

博客作为 Web 2.0 时代的重要应用之一，已经得到广泛地应用，所以 Web 开发人员应该掌握博客的基本开发流程以及博客中各个模块的具体功能和实现方式。本章的博客主要具有发表、查询、删除和分页浏览博客文章，发表、删除文章评论，查询、上传、删除分页浏览图片，添加、查询、删除、分页浏览朋友圈，以及查询、浏览和删除用户信息等功能。

注册为正式用户后，即可发表博客文章、评论博客文章、上传图片、设置自己的朋友圈。普通用户具有删除自己所写的文章和上传的图片的权限，网站管理员具有删除普通用户及普通用户所发表的文章和上传的照片的权限。博客管理系统的首页运行效果如图 25.1 所示。



图 25.1　博客管理系统首页运行效果

25.2 系 统 分 析

无论是简单的桌面应用还是复杂的电子商务网站，在进行项目开发前都需要花费大量的时间对系统各个方面进行分析，“磨刀不误砍柴工”，好的开头才是整个项目顺利、合格竣工的有效保障。

25.2.1 需求分析

在网络高速发展的今天，博客管理系统给人们提供了抒发个人情感、人与人之间进行良好沟通的平台。博客拥有真实的内容，人们可以通过博客记录工作、学习、生活和娱乐的点滴，以及发表文章和对事物的评论，从而在网上建立一个完全属于自己的天地，博客已成为当今网络最为个性化和平民化的个人展示的空间和平台。对于网民，只要拥有博客，就可以超越现实生活，拥有不同以往的全新网上生活；对于博客服务商，则必须从功能提供转型到全方位社会服务的提供，建立虚拟社会，并负责其维护，保证博客日常运行；对于社会而言，有利于构建和谐的互联网空间，维护和谐的社会环境。从这个角度来说，构建新生活方式，将是互联网发展的一个里程碑。很多门户网站都为注册用户提供



了免费的个人博客空间，也因此吸引了很多用户注册，从而很大程度上提高了网站的访问量和知名度，所以在开发社区性、娱乐性等网站中，提供博客模块是很有必要的。

25.2.2　系统目标

根据网站性质和需求，网站所包含的功能也是不同的，所以在项目开发前一定要认真做好需求分析，掌握用户所需和项目目标。本博客系统主要实现如下目标：

☑　系统采用 B/S 架构，实现人机交互。
☑　系统运行稳定，安全可靠。
☑　系统界面设计美观友好，操作简便。
☑　突出重点内容，显示最新文章和图片等信息。
☑　非登录用户可以浏览文章、浏览图片、发表评论，登录用户可以管理自己的文章和上传的图片等信息。
☑　全面的搜索查询功能，包括精确查询和模糊查询。
☑　完善的文章管理功能，包括文章的发表、删除，及对文章的评论与回复。
☑　支持图片上传功能。
☑　支持好友功能。
☑　支持公告栏功能。

25.2.3　系统功能结构

本系统的项目描述中已经详细介绍了该博客所具有的主要功能，为了使读者更加直观和理性地了解该博客系统所具有的功能，下面以结构图的形式进行展示，如图 25.2 所示。

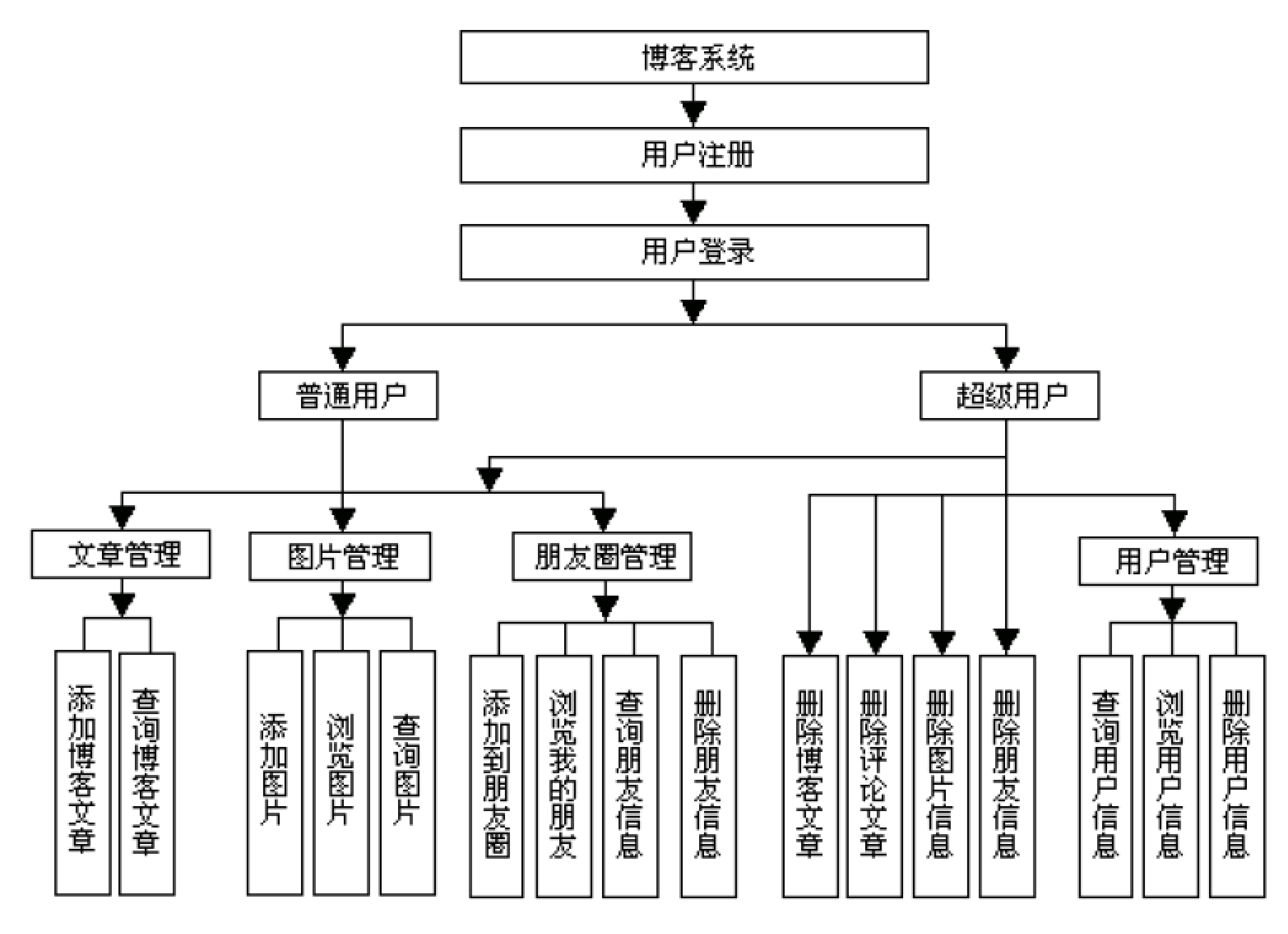


图 25.2　博客管理系统的功能结构图

25.2.4 系统流程图

完成系统的基本需求和功能分析后，即可规划出博客所具备的功能和操作流程。本博客系统的流程图如图 25.3 所示。

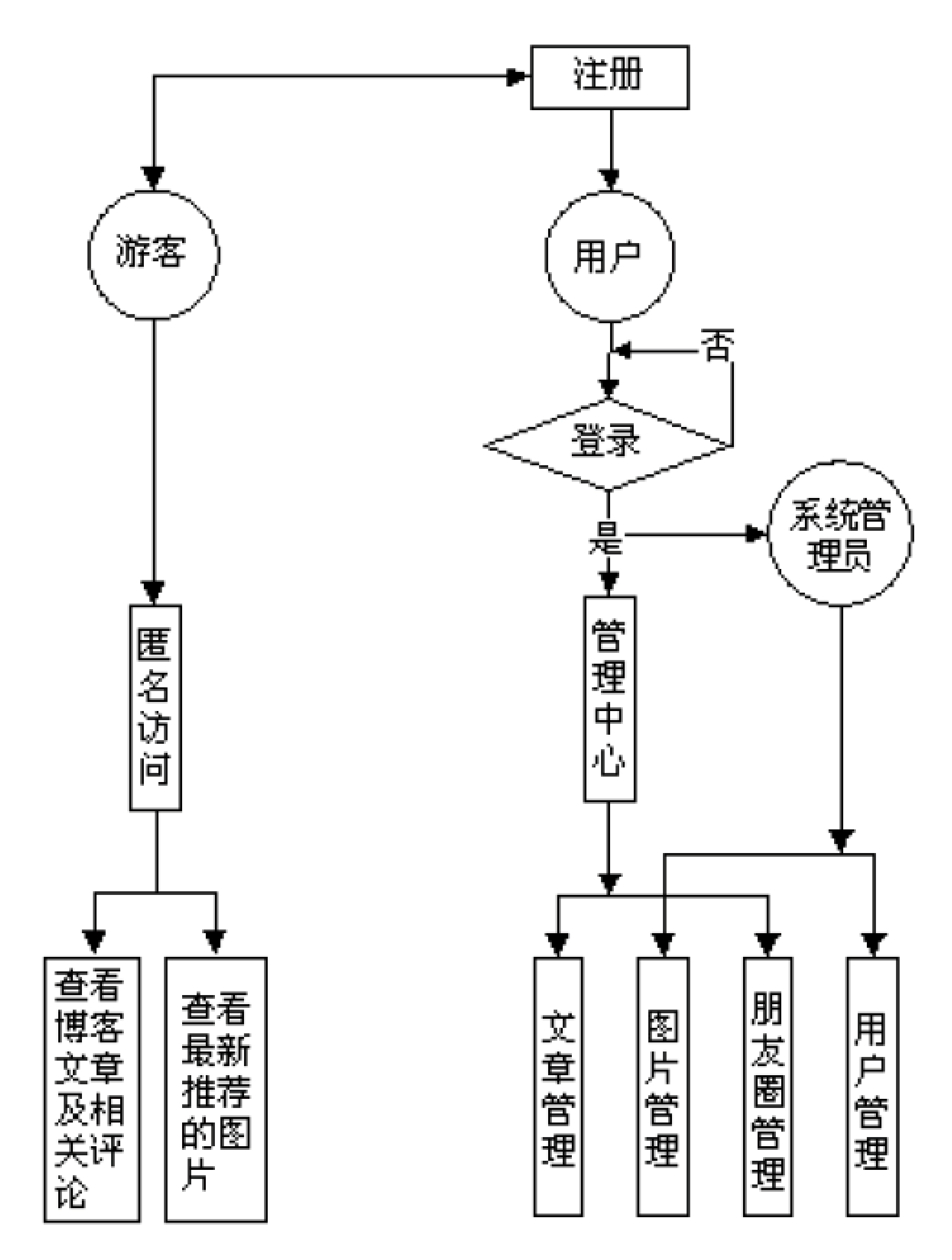


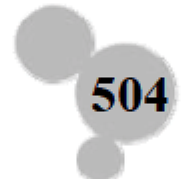
图 25.3 博客管理系统流程图

25.3 软件开发环境

项目能够顺利竣工，好的开发环境和开发工具也是必要的。在开发博客管理系统平台时，该项目使用的软件开发环境如下。

☑ 服务器端

- 操作系统：Windows Server 2003。
- 服务器：Apache 2.0。
- PHP 软件：PHP 5.0。
- 数据库：MySQL 5.0。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- 开发工具：Dreamweaver 8 和 Zend Studio for Eclipse。
- 浏览器：IE 6.0。
- 分辨率：最佳效果 1024×768 像素。



☑ 客户端
- 浏览器：推荐使用 IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

25.4　数据库与数据表的设计

目前，数据库是程序开发中大量数据有效存储的最佳载体，任何计算机语言都有与之结合性最好的数据库系统。PHP 一般与 MySQL 数据库组合开发，所以本博客系统也将采用 MySQL 作为后台数据库。下面具体讲解本系统中数据库及数据表的设计过程。

25.4.1　数据库分析

PHP 可以支持多种数据库，如 Oracle、Sybase、SQL Server、mSQL、MySQL 等。在这些数据库中，由于 MySQL 具有免费、跨平台、使用方便、访问效率较高等特点，所以得到广泛地应用，是业界公认的 PHP 的“黄金搭档”，Linux 系统、Apache 服务器、MySQL 数据库和 PHP 语言构成了现在主流的 LAMP 网站架构模式。综上原因，本博客系统也将使用 MySQL 作为博客系统中数据存储的载体。

25.4.2　数据库 E-R 图设计

根据需求分析和功能设计，本系统拟订了用户实体、上传图片实体、朋友圈实体、文章实体以及留言实体。为了使读者更清晰地了解每个实体所具备的功能，下面给出系统中主要实体的 E-R 模型。

用户是博客活动的主要参与者，必须对用户身份进行收集，所以将用户个人信息抽象为用户实体表示。用户实体包括注册用户的详细信息，如果想在本博客系统中进行发表文章、上传图片等操作，则必须要先进行注册。用户信息实体 E-R 图如图 25.4 所示。

博客可以保存用户上传的图片、照片等内容，为了有效地保存这些内容，可以将图片抽象为图片实体。图片实体主要包括上传图片的名称、上传图片的用户和上传图片时间等，图片实体 E-R 图如图 25.5 所示。

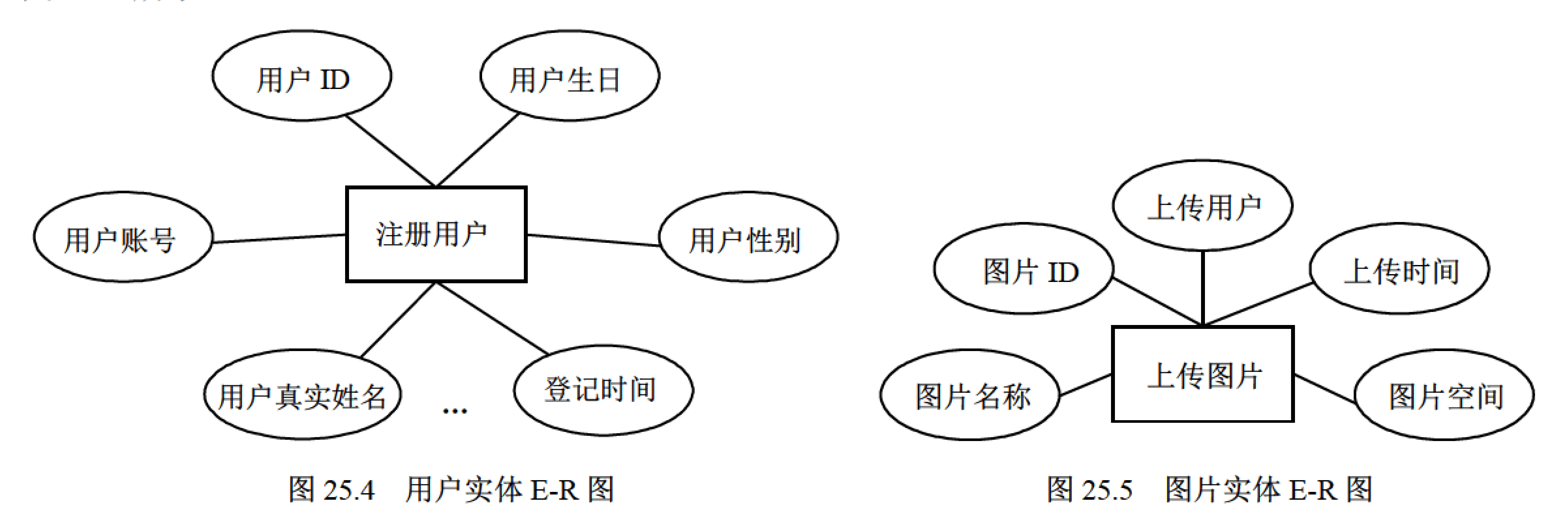


图 25.4　用户实体 E-R 图　　图 25.5　图片实体 E-R 图

25.4.3 创建数据库和数据表

完成对博客实体分析后，下面根据分析结果创建数据库和数据表。创建和维护 MySQL 数据库及数据表，既可以在 Windows 系统的命令提示符窗口或 Linux/UNIX 系统的终端命令窗口中实现，也可以使用 MySQL 的图形化界面管理工具实现。MySQL 图形界面管理工具有多种，其中，phpMyAdmin 是由 PHP 编写的通过客户端浏览器对 MySQL 数据库进行管理的工具，其完全免费、安装方便，所以本系统将使用 phpMyAdmin 作为 MySQL 数据库创建和维护的工具。

在浏览器中打开 phpMyAdmin，如图 25.6 所示，在“创建一个新的数据库”选项组中的文本框中输入“db_database25”作为博客数据库的名称，然后单击“创建”按钮即可完成数据库的创建。

完成数据库的创建后，即可创建博客所要使用的数据表。根据实体 E-R 图和本系统的实际情况，需要创建 6 张数据表，其中 db_database25 数据库中所包含的表如图 25.7 所示。



图 25.6 创建数据库

下面具体介绍这 6 张数据表的结构设计和说明。

☑ tb_article（文章表）

文章表用于存储用户发表过的文章信息。tb_article 表的结构如图 25.8 所示。



服务器: localhost ▸ 数据库: db_database25

结构 SQL 搜索 查询 导出 Import 操作 权限 删除

表	记录	类型	整理	大小	多余
tb_article	4	MyISAM	gb2312_chinese_ci	3.5 KB	-
tb_filecomment	2	MyISAM	gb2312_chinese_ci	2.1 KB	-
tb_friend	5	MyISAM	gb2312_chinese_ci	2.4 KB	-
tb_public	1	MyISAM	gb2312_chinese_ci	2.1 KB	-
tb_tpsc	3	MyISAM	gb2312_chinese_ci	160.4 KB	-
tb_user	3	MyISAM	gb2312_chinese_ci	2.2 KB	-
6 个表 总计	18	MyISAM	gb2312_chinese_ci	172.8 KB	0 字节

图 25.7 db_database25 数据库所包含的数据表

服务器: localhost ▸ 数据库: db_database25 ▸ 表 : tb_article

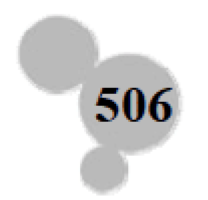
浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外
id	int(10)			否		auto_increment
content	mediumtext	gb2312_chinese_ci		否		
author	varchar(20)	gb2312_chinese_ci		否		
now	datetime			否	0000-00-00 00:00:00	
title	varchar(200)	gb2312_chinese_ci		否		

图 25.8 文章表结构

☑ tb_filecomment（评论表）

评论表存储的是用户对文章的评论，注册用户和游客都可以发表评论。tb_filecomment 表的结构如图 25.9 所示。



☑　tb_friend（好友信息表）

好友信息表主要记录好友的姓名、性别、生日等好友的个人信息。tb_friend 表的结构如图 25.10 所示。

服务器: localhost ▸ 数据库: db_database25 ▸ 表 : tb_filecomment

字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
fileid	int(4)			否	0	
username	varchar(20)	gb2312_chinese_ci		否		
content	text	gb2312_chinese_ci		否		
datetime	datetime			否	0000-00-00 00:00:00	

图 25.9　评论表结构



服务器: localhost ▸ 数据库: db_database25 ▸ 表 : tb_friend

字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
name	varchar(50)	gb2312_chinese_ci		否		
sex	varchar(10)	gb2312_chinese_ci		否		
bir	date			否	0000-00-00	
city	varchar(50)	gb2312_chinese_ci		否		
address	varchar(100)	gb2312_chinese_ci		否		
postcode	varchar(6)	gb2312_chinese_ci		否		
email	varchar(50)	gb2312_chinese_ci		否		
tel	varchar(20)	gb2312_chinese_ci		否		
handset	varchar(20)	gb2312_chinese_ci		否	0	
QQ	varchar(20)	gb2312_chinese_ci		否	0	
username	varchar(20)	gb2312_chinese_ci		否		

图 25.10　好友信息表结构

☑　tb_public（公告栏信息表）

公告栏信息表主要记录了网站情况、博客系统的版本情况或网站活动等内容。公告栏信息表的结构如图 25.11 所示。

☑　tb_tpsc（图片信息表）

图片信息表存储的是上传图片的信息，如图片名称、上传用户、上传时间等。tb_tpsc 表的结构如图 25.12 所示。



服务器: localhost ▸ 数据库: db_database25 ▸ 表 : tb_public

字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
title	varchar(50)	gb2312_chinese_ci		否		
content	varchar(200)	gb2312_chinese_ci		否		
pub_time	date			否		

图 25.11　公告栏信息表结构



服务器: localhost ▸ 数据库: db_database25 ▸ 表 : tb_tpsc

字段	类型	整理	属性	Null	默认	额外
id	int(10)			否		auto_increment
tpmc	varchar(30)	gb2312_chinese_ci		否		
file	blob		BINARY	否		
author	varchar(20)	gb2312_chinese_ci		否		
scsj	date			否	0000-00-00	

图 25.12　图片信息表结构

☑　tb_user（用户信息表）

用户信息表主要存储用户的个人信息。tb_user 表的结构如图 25.13 所示。

服务器: localhost ▸ 数据库: db_database25 ▸ 表 : tb_user

字段	类型	整理	属性	Null	默认	额外
id	int(20)			否		auto_increment
regname	varchar(20)	gb2312_chinese_ci		否		
regrealname	varchar(20)	gb2312_chinese_ci		否		
regpwd	varchar(40)	gb2312_chinese_ci		否		
regbirthday	date			否	0000-00-00	
regemail	varchar(100)	gb2312_chinese_ci		否		
regcity	varchar(100)	gb2312_chinese_ci		否		
regico	varchar(50)	gb2312_chinese_ci		否		
regsex	varchar(4)	gb2312_chinese_ci		否		
regqq	varchar(40)	gb2312_chinese_ci		否		
reghomepage	varchar(100)	gb2312_chinese_ci		否		
regsign	varchar(200)	gb2312_chinese_ci		否		
regintroduce	mediumtext	gb2312_chinese_ci		否		
ip	varchar(20)	gb2312_chinese_ci		否		
fig	int(1)			否	0	

图 25.13　用户信息表结构

25.5 首页设计

首页是整个博客的入口地址，合理的首页布局和模块分配直接影响到用户对博客的第一印象。为了吸引更多的用户浏览和使用博客，在开发项目时需要精心地对首页进行设计。

25.5.1 首页概述

本博客系统在页面设计上采用传统的排版方式，主要由首部导航栏、左侧显示区和主显示区 3 部分组成，如图 25.14 所示。

- ☑ 首部导航栏：包括导航链接、用户登录等模块。
- ☑ 左侧显示区：包括最新文章、最新图片和系统时间等模块。
- ☑ 主显示区：为系统公告栏，显示系统及网站的最新咨询等信息。

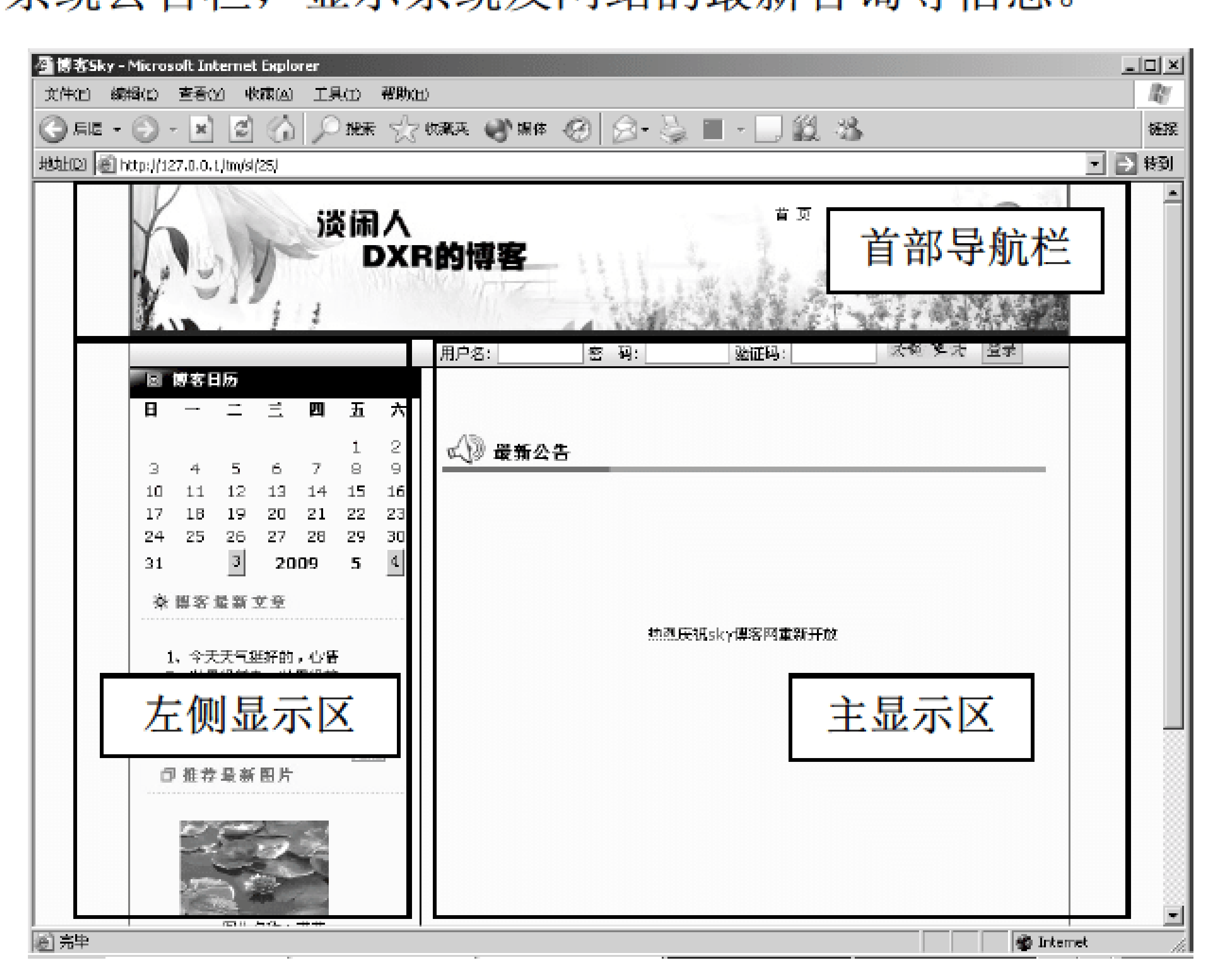


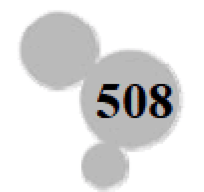
图 25.14　博客系统首页结构

25.5.2 首页技术分析

本博客首页的主显示区是一个可以自下而上移动的公告栏。公告栏主要用于公布系统版本的更新或升级信息、网站的最新活动安排等内容，也可以链接一些用户的精彩文章。本系统的公告栏可以上下移动，主要是应用 HTML 语言的<marquee>标记实现。使用<marquee>标签可以实现文字或图片的滚动效果，增强了实用性的同时，也增强了观赏性。下面将介绍这个标签的常用属性和方法。

1. <marquee>标签的文字移动属性

<marquee>标签的特点就是可以使文字或图片上下移动。早期，这是一个了不起的技术，随着 W3C 标准的逐渐完善，<marquee>标签已经少有用武之地了。但有些技术是永远不变的，如对于<marquee>



而判断用户所输入的验证码是否正确，代码如下。

```
<?php
$str=array("大","更","创","天","科","客","博","技","立","新");          //汉字数组
$word=strlen($str);
for($i=0;$i<4;$i++){
    $num=rand(0,$word*2-1);                                         //随机获取汉字数组的键值
    $img=$img."<img src=' images/checkcode/".$num.".gif' width='16' height='16'>";     //显示随机图片
    $pic=$pic.$str[$num];                                           //将图片转换成数组中的文字
}
?>
```

25.5.3 首页的实现过程

视频讲解：光盘\TM\Video\25\首页的实现过程.exe

（1）使用 session_start()函数启用 Session 支持，并使用 include 语句包含数据库连接文件，从而在首页中导入与数据库的连接代码。

```
<?php
session_start();                        //开启 Session 功能
include "Conn/conn.php";                //包含数据库连接文件
?>
```

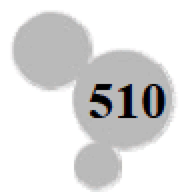
（2）建立用户登录表单，用于录入用户的登录信息，表单及表单结构如表 25.3 所示。

表 25.3 表单及表单元素

元素类型	元素名称	属性设置	说明
表单	form	name="form" method="post" action="checkuser.php"	表单
文本域	txt_user	type="text" name="txt_user" size="10"	用户名
密码域	txt_pwd	name="txt_pwd" type="password" style="FONT-SIZE: 9pt; WIDTH: 65px" size="6"	密码
文本域	txt_yan	name="txt_yan" style="FONT-SIZE: 9pt; WIDTH: 65px" size="8"	输入验证码
隐藏域	txt_hyan	type="hidden" name="txt_hyan" id="txt_hyan" value="<?php echo $pic;?>"	保存随机生成的验证码内容
提交按钮	sub_dl	style="FONT-SIZE: 9pt" type="submit" value="登录" name="sub_dl" onClick="return f_check(form)"	提交按钮

（3）在左侧显示区内显示最新的 5 篇文章的主题，代码如下。

```
<?php
    $sql=mysql_query("select id,title from tb_article order by now desc limit 5");     //查询最新 5 篇文章
    $i=1;
    while($info=mysql_fetch_array($sql))        //循环显示文章主题
    {
?>
    <tr>
        <td  width="201"  align="left"  valign="top">   <a  href="article.php?file_id=<?php  echo
```



```
$sql=mysql_query("select * from tb_user where regname = '$UserName'");
$result=mysql_fetch_array($sql);
if ($result!=false){                              //如果用户名已经被注册则给出提示
    echo ("<script>alert('用户名已被注册！');history.go(-1);</script>");
    exit();
}
$_SESSION[username]=$_POST[txt_regname];       //将用户名保存到 Session 中
$regname=$_POST[txt_regname];                  //接收表单内容
$regrealname=$_POST[txt_regrealname];
$regpwd=md5($_POST[txt_regpwd]);
$regbirthday=$_POST[txt_birthday];
$regemail=$_POST[txt_regemail];
$regcity=$_POST[txt_province].$_POST[txt_city];
$regico=$_POST[txt_ico];
$regsex=$_POST[txt_regsex];
$regqq=$_POST[txt_regqq];
$reghomepage=$_POST[txt_reghomepage];
$regsign=$_POST[txt_regsign];
$regintroduce=$_POST[txt_regintroduce];
$ip=getenv(REMOTE_ADDR);                          //获取用户 IP
$INS=mysql_query("Insert Into tb_user (regname, regrealname, regpwd, regbirthday, regemail, regcity, regico, regsex, regqq, reghomepage, regsign, regintroduce, ip, fig) Values ('$regname', '$regrealname', '$regpwd', '$regbirthday', '$regemail', '$regcity', '$regico', '$regsex', '$regqq', '$reghomepage', '$regsign', '$regintroduce', '$ip', 0)");
echo "<script> alert('用户注册成功！');</script>";          //提示用户注册成功
echo "<script> window.location='file.php';</script>";     //重新定位到 file.php 页面
?>
```

25.6.3 用户登录

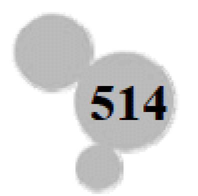
视频讲解：光盘\TM\Video\25\用户登录.exe

如果用户已经注册为本博客的会员，那么在发表文章或上传图片时需要进行登录。本博客的登录页面如图 25.16 所示。



图 25.16 用户登录页面

在介绍首页的实现过程时，已经详细介绍了用户登录表单及表单元素，这里不再赘述。当用户填写完个人信息后，单击“登录”按钮即可将用户录入的表单信息提交到 chkuser.php 页面，该页面用于对用户的登录信息进行验证，代码如下。



JavaScript 在前台对用户所输入的内容进行验证，代码如下。

```
<script language="javascript">
function check(form){                                          //定义 check()函数
      if (document.myform.sel_key.value==""){                  //判断关键字是否为空
            alert("请输入查询条件!");myform.sel_key.focus();return false;
      }
}
</script>
```

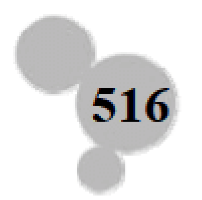
（3）用户正确填写查询表单中的内容并单击“提交”按钮后，查询信息将被提交到当前页进行处理，并将查询结果显示在当前页，代码如下。

```
<?php
if (isset($_POST[sel_key])){                                   //判断是否单击“提交”按钮
    $tj=$_POST[sel_tj];                                        //接收提交的查询信息
    $key=$_POST[sel_key];
    $sql=mysql_query("select * from tb_user where $tj='$key'");  //查询用户信息
    $result=mysql_fetch_array($sql);                           //获取查询结果集
    if($result==false){
          echo ("[<font color=red>Sorry!该用户不存在!</font>]");
    } else{
?>
<!--省略了部分代码-->
<?php
               }
       }                                                       //查询模块结束
?>
```

进入用户信息查询模块，首先选择查询条件并在“关键字”文本框中输入要查询的关键字，然后单击“检索”按钮将显示出查询结果。例如，查询条件选择“用户名”，关键字输入“Tsoft”，然后单击“检索”按钮，将在页面中显示名为 Tsoft 的用户信息，如图 25.17 所示。



图 25.17　查询用户信息



```
                <td align="right" bgcolor="#E9EAE5" class="hongse01">
<?php
            if($page!=1){
                echo   "<a href=browseuser.php?page=1>首页</a> ";
                echo "<a href=browseuser.php?page=".($page-1).">上一页</a> ";
            }
            if($page<$page_count)
            {
                echo "<a href=browseuser.php?page=".($page+1).">下一页</a> ";
                echo   "<a href=browseuser.php?page=".$page_count.">尾页</a>";
            }
        }
}
?>
```

进入用户信息浏览模块，如图 25.18 所示，用户可以通过单击分页链接实现用户信息的分页显示。如果是管理员登录，还可以删除指定的用户信息。



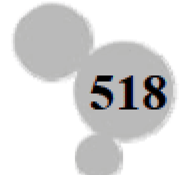
图 25.18　浏览用户信息

25.7　文章管理模块设计

文章管理模块是博客系统中较为重要的模块之一，因为博客的一个核心功能就是为用户提供发表自己的文章的功能。所以，对博客的文章管理模块进行精心设计是必要的。

25.7.1　文章管理模块概述

本系统的文章管理模块包括“发表文章”、“查看文章”、“其他用户文章”、“发表评论”、“删除文章”和“删除评论”6 项功能。其中，普通用户只能删除自己的文章及对文章进行评论，只有管理员才有权删除任何一篇文章及回复。文章管理模块的框架如图 25.19 所示。



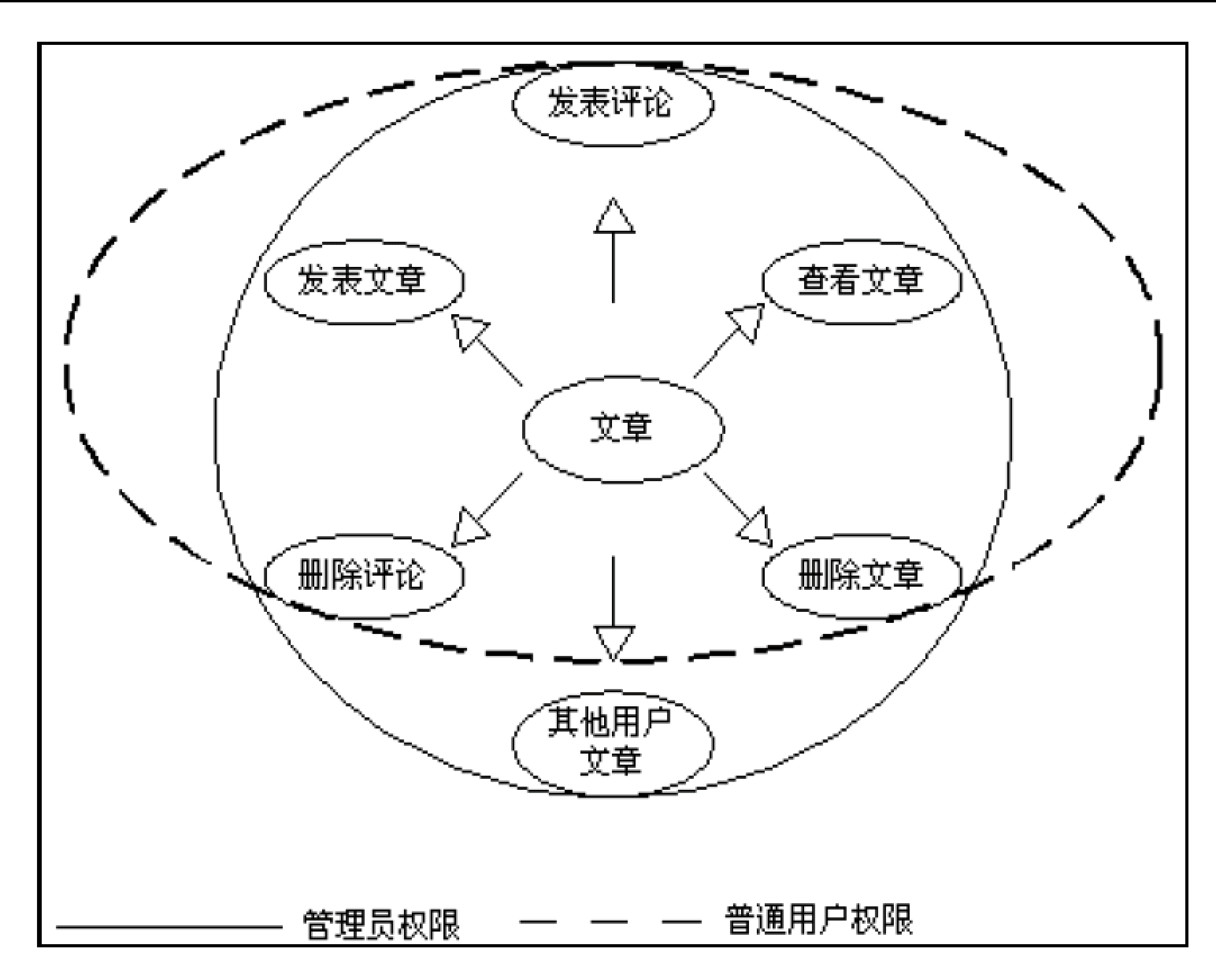


图 25.19　文章管理模块框架图

25.7.2　文章管理模块技术分析

视频讲解：光盘\TM\Video\25\文章管理模块分析.exe

大多数情况下，只有用户成功登录后才可以在论坛或博客中发表帖子或文章，这样用户可以有效地管理自己所发布的内容，而且可以定位不同文章的作者。25.7.1 节中已经介绍了不同用户具有不同的权限，那么如何为用户分配权限呢？有 Web 编程基础的读者都知道通过 Session 可以跟踪和定位访问 Web 服务器的不同用户，同样，PHP 也可以采用 Session 来保存用户的身份信息，在进行站内活动时，根据 Session 所保存的用户身份信息来判断用户是否具备某项操作的权限。

Session 的中文含义是“会话”，是指用户从进入网站开始，直到关闭网站这段时间内，所有网页共同使用的公共变量的存储机制。Session 比 Cookie 更有优势：Session 是存储在服务器端的，不易被伪造；Session 的存储没有长度限制；Session 的控制更容易、更安全。

程序开发人员要在程序编写过程中灵活运用 Session，首先应该掌握 Session 的相关配置，PHP 中对 Session 进行配置是在 php.ini 文件中进行的。表 25.7 为 PHP 中 Session 的相关配置。

表 25.7　Session 的常用配置选项

配 置 选 项	说　　明
session.save_path = c:/temp	保存 Session 变量的目录，在 Linux/UNIX 下为/tmp
session.ues_cookies = 1	是否使用 Cookie
session.name = PHPSESSID	表示会话 ID
session.auto_start = 0	是否自动启用 Session，当为 1 时，在每页中不必调用 Session_satrt()函数
session.cookie_lifetime = 0	设定 Cookie 送到浏览器后的保存时间，单位为秒。默认值为 0，表示直到浏览器关闭
session.cookie_path = /	Cookie 有效路径
session.cookie_domain =	有效域名

尽可能的将 session_start()函数放到第一行。

25.7.3　添加文章的实现过程

视频讲解：光盘\TM\Video\25\添加文章实现过程.exe

用户成功登录博客后，会自动跳转到添加文章页面，当然也可以通过单击“文章管理”/“添加博客文章”超链接返回到添加文章页面。文章添加页面的运行结果如图 25.20 所示。



图 25.20　添加文章页面的运行结果

添加文章页面所涉及的表单及表单元素如表 25.9 所示。

表 25.9　添加文章页面中的表单及表单元素

元素类型	元素名称	属性设置	含　义
form	myform	method="post" action="check_file.php"	添加文章表单
text	txt_title	id="txt_title" size="68"	文章标题
select	font	name="font" class="wenbenkuang" id="font" onChange="showfont(this.options[this.selectedIndex].value)"	文章字体
select	size	class="wenbenkuang" onChange="showsize(this.options[this.selectedIndex].value)"	字体大小
select	color	onChange="showcolor(this.options[this.selectedIndex].value)" name="color" size="1" class="wenbenkuang" id="select"	字体颜色
textarea	file	cols="75" rows="20" id="file" style="border:0px;width:520px;"	文章内容
submit	btn_tj	id="btn_tj" value="提交" onClick="return check();"	提交按钮

用户填写完博客主题和文章内容后，单击“提交”按钮，系统将跳转到处理页（check_file.php）进行处理。在处理页中，将传过来的文章标题、文章作者和文章内容等参数组成 insert 语句，并最终保存到数据表中。如果添加信息成功，系统返回到本页，可继续执行添加操作；如果添加失败，则返回到上一步，程序的关键代码如下：

```
<?php
session_start();                                        //开启 Session 支持
include "Conn/conn.php";                                //包含数据库连接文件
if($btn_tj<>""){                                        //判断传值页面
    $title=$_POST[txt_title];                           //取得文章标题
    $author=$_SESSION[username];                        //取得作者
    $content=$_POST[file];                              //取得文章内容
    $now=date("Y-m-d H:i:s");                           //使用 date 函数生成发布时间
    /*  生成 insert 语句  */
    $sql="Insert Into tb_article (title,content,author,now) Values ('$title','$content','$author','$now')";
    $result=mysql_query($sql);                          //执行 insert 语句
    /*  根据$result，返回结果  */
    if($result){
        echo "<script>alert('恭喜您，您的文章发表成功!!!');window.location.href='file.php';</script>";
    }else{
        echo "<script>alert('对不起，添加操作失败!!!');history.go(-1);</script>";
    }
}else{
    echo "<script>alert('对不起，添加操作失败!!!');history.go(-1);</script>";
}
?>
```

25.7.4 文章列表的实现过程

视频讲解：光盘\TM\Video\25\文章列表实现过程.exe

单击“文章管理”/“我的文章”超链接，将显示用户发表过的文章列表，文章列表页面（myfiles.php）的运行结果如图 25.21 所示。



图 25.21　文章列表页面的运行结果

文章列表页面使用了分页技术和 do…while 循环语句来输出文章标题。程序关键代码如下：

```
<?php
session_start();                                        //开启 Session 支持
include "Conn/conn.php";                                //包含数据库连接文件
include "check_login.php";                              //包含权限检查文件
?>
<!--   分页  -->
<?php
/*   $page 为当前页，如果$page 为空，则初始化为 1    */
```



25.8.2 图片上传模块技术分析

视频讲解：光盘\TM\Video\25\图片上传模块技术分析.exe

博客中，用户除了可以发表文章，还可以将自己的照片或喜爱的图片发布到网上，保存到自己的博客中。文字信息可以通过简单的表单保存到数据库中，那么图片该如何从本地保存到服务器中呢？与保存文件类似，图片可以通过上传的方式保存到服务器某目录中，也可以转换为二进制数据保存到数据库中。

1．上传图片的基本流程

上传图片到服务器既可以通过 FTP 实现，也可以通过 POST 方法实现。本系统中使用 POST 方法实现，采用这种方法更加方便，具体实现步骤如下：

（1）在<form>表单中定义文件选择域，用来选择要上传的图片。这里要注意，通过 POST 方法上传文件或图片时，一定要在 form 表单中加上属性 enctype="multipart/form-data"，否则上传不了文件（图片）。

（2）在处理文件上传的页面中，使用$_FILES 全局数组来判断上传文件类型和上传文件（图片）大小是否符合要求。

$_FILES 数组为系统预定义的全局数组，保存的是上传文件（图片）的相关属性。使用格式为：

```
$_FILES[name][property];
```

其中，第一维的键值 name 要与上传表单中的文件选择域名相同，第二维的键值 property 的取值说明如表 25.10 所示。

表 25.10 $_FILES 的相关属性

属　性　值	说　　明
type	上传文件的类型
size	上传文件的大小
tmp_name	上传文件在服务器中的临时文件名
error	上传文件失败的错误代码

（3）使用 move_uploaded_file()函数上传文件（图片）。

（4）返回页面等待下一步操作。

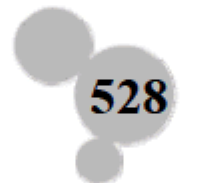
2．使用 POST 方法上传图片

PHP 能够接受任何来自符合 RFC-1867 标准的浏览器（包括 Netscape Navigator 3 及更高版本；Microsoft Internet Explorer 3 加微软补丁，或者更高版本）上传的文件。PHP 的这种特性使用户既可以上传文本文件，也可以上传二进制文件。

下面具体讲解如何使用 POST 方法上传图片或文件。

利用 move_uploaded_file()函数上传文件，首先应在前台表单中做如下处理：

```
<form enctype="multipart/form-data" action="URL" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="30000">
```



```
上传文件路径：  <input name="file" type="file">
<input type="submit" value="上传">
</form>
```

上述代码中的 URL 应该替换成指向一个 PHP 文件的真实提交地址，在该文件中将使用 move_uploaded_file()函数进行文件上传。

取值为 MAX_FILE_SIZE 的隐藏域（单位为字节）必须先于文件选择框，其值为可以接收的文件的最大尺寸。同时，要保证文件上传表单中有 enctype="multipart/form-data"的属性，否则文件上传将不能实现。

MAX_FILE_SIZE 的值只是对浏览器的一个建议，实际上可以不进行设置，php.ini 文件中已经设置了允许上传文件的最大值。但最好还是在表单中加上 MAX_FILE_SIZE，因为它可以避免用户在长时间等待大文件上传之后才发现该文件过大的麻烦。

为上传文件定义的变量会根据 PHP 的版本及设置的不同而存在差异。自动全局变量$_FILES 从 PHP 4.1.0 版本开始被支持。在这之前，从 4.0.0 版本开始，PHP 支持$HTTP_POST_FILES 数组。这些数组将包含所有关于用户上传的文件的信息，这里推荐使用$_FILES。如果 PHP 的设置选项 register_globals 为 On，则相关的变量名也将会存在。从 PHP 4.2.0 版本开始，register_globals 的默认值被设为 Off。

$_FILES 数组的内容如下所示（假设文件上传字段的名称为 file）。

```
$_FILES ["file"]["name"]
```

表示客户端机器文件的原名称。

```
$_FILES["file"]["type"]
```

表示文件的 MIME 类型，需要浏览器提供该信息的支持，如“image/gif”。

```
$_FILES["file"]["size"]
```

表示已上传文件的大小，单位为字节。

```
$_FILES ["file"]["tmp_name"]
```

表示文件被上传后在服务器端储存的临时文件名。

下面根据一段代码介绍如何使用函数 move_uploaded_file()进行文件上传，如下所示。

```
function getname($exname) {
    $dir = "upimages/";                        //定义上传文件的存储路径
    $i=1;
    if(!is_dir($dir)){
        mkdir($dir,0777);                      //判断定义的路径是否已经存在，如不存在则创建它
    }
    while(true){
        if(!is_file($dir.$i.".".$exname)){
            $name=$i.".".$exname;
            break;        //用数字加扩展名作为上传文件的存储名，如果该文件已经存在，则使数字加 1 后再存储
        }
        $i++;
```

```
        }
    return $dir.$name;
}
$exname=strtolower(substr($_FILES['upfile']['name'],(strrpos($_FILES['upfile']['name'],'.')+1)));
                                                    //获取上传文件的扩展名
$uploadfile = getname($exname);
move_uploaded_file($_FILES['upfile']['tmp_name'], $uploadfile);
```

上述代码中主要应用到函数 is_dir()、mkdir()、is_file()、strtolower()、strrpos()、move_uploaded_file()，下面将逐一介绍这些函数。

☑ is_dir()函数

```
bool is_dir(string filename)
```

is_dir()函数的作用是判断该函数的参数是否是一个目录，如果该参数存在而且是一个目录就返回 true，否则返回 false。如应用 is_dir()函数判断“c:/mr”是否为目录的代码如下：

```
<?php
  if(is_dir("c:/mr")){                    //判断 c:/mr 是否为目录
      echo "是目录!";
  }else{
      echo "不是目录!";
  }
?>
```

☑ mkdir()函数

```
bool mkdir ( string pathname [, int mode])
```

mkdir()函数的作用是创建由 pathname 指定的目录。mode 在 Windows 下被忽略，自 PHP 4.2.0 起为可选项。默认的 mode 为 0777，表示最大可能的访问权。如应用 mkdir()函数在当前目录下建立目录 mr，代码如下：

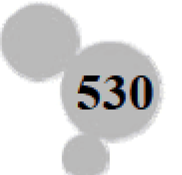
```
<?php
  if(mkdir("mr")){                    //创建目录 mr
       echo "目录建立成功!";
  }else {
     echo "目录建立失败!";
  }
?>
```

☑ is_file()函数

```
bool is_file(string filename)
```

is_file()函数的作用是判断 filename 指定的文件是否为一个常规文件，如果该文件存在而且是常规文件就返回 true，否则返回 false。如应用 is_file()函数判断“c:/mr/mr.txt”是否为合法文件，代码如下：

```
<?php
  if(is_ file("c:/mr/mr.txt")){      //判断 c:/mr/mr.txt 是否为一个合法文件
```



```
    echo "是合法文件!";
  }else{
    echo "不是合法文件!";
  }
?>
```

☑　strtolower()函数

```
string strtolower(string str)
```

strtolower()函数的作用是将 str 中的字母全部转换成小写字母。如应用 strtolower()函数将字符串中的大写字母转换成小写字母。其代码如下：

```
<?php
echo strtolower("PHP 函数参考手册");
echo strtolower("JAVA 案例精选");
?>
```

本示例的运行结果如下：

php 函数参考手册

java 案例精选

☑　strrpos()函数

```
int strrpos(string str1,char str2)
```

strrpos()函数的作用是在 str1 中查找 str2 首字符最后一次出现的位置。如应用 strrpos()函数查找指定字符在字符串中最后出现的位置，代码如下：

```
<?php
echo strrpos("Hello world!","WO");       //区分大小写，搜索失败
echo strrpos("I am fine !","f",5);        //从第 5 个字符开始搜索，输出 5
echo strrpos("明日图书","图书");           //输出 4
?>
```

本示例的运行结果为：54

☑　move_uploaded_file()函数

```
bool move_uploaded_file ( string filename, string destination)
```

move_uploaded_file()函数的作用是将 filename 指定的文件上传到 destination 指定的位置。如应用 move_uploaded_file()函数将“c:/mr/mr.rar”上传到服务器当前目录下的 mrsoft 目录中，代码如下：

```
<?php
  if(!is_dir("mrsoft")){
    mkdir("mrsoft");
  }
  if(move_uploaded_file("c:/mr/mr.rar","mrsoft/mr.rar")){
    echo "文件上传成功!";
  }else{
   echo "文件上传失败!";
```

表 25.11　图片上传页面中的表单元素

元素类型	元素名称	属性设置	含　义
form	myform	method="post" action="tptj_ok.php"　enctype="multipart/form-data"	图片上传表单
text	tpmc	type="text" id="tpmc" size="40"	图片名称
file	file	type="file" size="23" maxlength="60"	上传文件域
submit	btn_tj	type="submit" id="btn_tj" value="提交" onClick="return pic_chk();"	提交按钮

当用户输入图片名称，并选择图片路径后，单击“提交”按钮，系统将进入到上传处理页（tptj_ok.php）中进行处理。在处理页中，首先对图片名称进行处理，去掉特殊字符、空行和空格，然后对上传的文件进行类型检查、文件大小检查。最后以二进制的形式与图片的其他信息（如上传用户、上传时间等）一起存进数据表中。关键代码如下：

```
<?php
session_start();                                    //开启 Session 支持
include "check_login.php";                          //包含权限检查文件
include "Conn/conn.php";                            //包含数据库连接文件
if($_POST["btn_tj"]=="提交"){
    $tpmc=htmlspecialchars($tpmc);                  //将图片名称中的特殊字符转换成 HTML 格式
    $tpmc=str_replace("\n","<br>",$tpmc);           //将图片名称中的回车符以自动换行符取代
    $tpmc=str_replace(""," ",$tpmc);           //将图片名称中的空格以“ ”取代
    $author=$_SESSION[username];
    $scsj=date("Y-m-d");                            //设置图片的上传时间
    $profix = array(".jpg",".gif",".jpeg");         //设置允许上传的文件后缀类型
    $f_name = $_FILES[file][name];                  //取得要上传的文件名
    $pro_name=substr($f_name,strrpos($f_name,"."));  //取得上传文件的后缀
     /* 判断上传文件的类型是否为允许类型 */
    if(!in_array(strtolower($pro_name), $profix)){
        echo "<script>alert('文件格式不对');history.go(-1);</script>";
        exit();
     }
     /* 判断上传文件的大小，如果文件过大，提示错误 */
     if($_FILES[file][size] > 500000){
         echo "<script>alert('文件上传错误,请重新上传');history.go(-1)</script>";
         exit();
     }else{
         $fp=fopen($file,"r");                      //以只读方式打开文件
         $file=addslashes(fread($fp,filesize($file)));   //将文件中的引号部分加上反斜线
         $query="insert into tb_tpsc (tpmc,file,author,scsj) values ('$tpmc','$file','$author','$scsj')";
         $result=mysql_query($query);
         echo "<meta http-equiv=\"refresh\" content=\"1;url=browse_pic.php\">图片上传成功，请稍等...";
     }
}
?>
```

上述代码中，主要涉及 htmlspecialchars()函数、str_replace()函数的使用方法，这两个函数在处理字符串时经常会用到，下面具体介绍这两个函数的使用方法。

☑ htmlspecialchars ()函数

```
string htmlspecialchars(string string)
```

该函数将一些特殊字符转换成 HTML 格式，而不会将所有字符都转换成 HTML 格式。该函数转换的特殊字符如表 25.12 所示。

表 25.12 htmlspecialchars()函数转换的特殊字符

参　　数	说　　明
&（和）	转成&
“ “” ”（双引号）	转成"
>（大于号）	转成>
<（小于号）	转成<

例如，应用 htmlspecialchars()函数将特殊字符转换成 HTML 格式，代码如下：

```
$str='<table width="371" border="0">特殊字符的转换</table>';
echo htmlspecialchars($str);
```

本示例的运行结果如下：

<table width="371" border="0">特殊字符的转换</table>

☑ str_replace ()函数

```
mixed str_replace ( mixed search, mixed replace, mixed subject , int &count )
```

str_replace()函数将所有在参数 subject 中出现的 search 以参数 replace 替换，参数&count 表示替换字符串执行的次数。

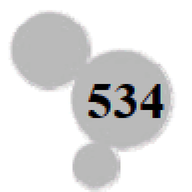
str_replace()函数的参数说明如表 25.13 所示。

表 25.13 str_replace()函数的参数说明

参　　数	说　　明
search	指定将要被替换的字符
replace	指定替换所使用的字符
subject	指定被操作的字符串
&count	替换字符串执行的次数

例如，应用 str_replace()函数实现对字符串的替换，代码如下：

```
<?php
$str = "吉林省明日科技";
$str1 = array("明日软件", "明日图书", "明日论坛");
$str2 = array("明日");
$strs = str_replace($str2, $str, $str1);
print_r($str1);                              //输出替换前的结果
print_r($strs);                              //输出替换后的结果
$str = str_replace("mrsoft", "吉林省明日科技有限公司", "Welcome to mrsoft!", $count);
echo "Welcome to mrsoft!";
```



```
echo $str;                                    //输出替换后的结果
echo $count;                                  //输出替换字符的次数
?>
```

25.8.4　图片浏览的实现过程

视频讲解：光盘\TM\Video\25\图片浏览的实现过程.exe

本页的实现代码与查看文章页面略有不同，查看文章页面中，每条数据占了一行，而查看图片则采用的是分栏显示，以每行两张图片的格式输出，每页显示 4 张图片。通过单击小图片，可以查看图片原效果图。图片浏览页面运行效果如图 25.25 所示。



图 25.25　浏览图片页面的运行结果

实现图片浏览的关键代码如下：

```
<?php
session_start();                                        //开启 Session 支持
include "Conn/conn.php";                                //包含数据库连接文件
include "check_login.php";                              //包含权限检查文件
?>
<!--首部导航栏 -->
<tr>
    <td height="192" align="center" valign="top" >
    <?php
    //分页显示
    $query="select * from tb_tpsc where scsj order by id desc limit $offset, $page_size";
    /*  返回结果集  */
    $result=mysql_query($query);
    ?>
    <table width="496" border="1" align="center" cellpadding="3" cellspacing="1" bordercolor="#D6D7D6">
        <tr>
        <?php
        /*  设置一个变量，判断当前数据是奇数输出还是偶数输出  */
        $i=1;
```

```
"52" height="16" alt="删除图片" onClick="return pic_chk();"></a>
                    <?php
                         }
                    ?>
               </td>
          </tr>
     </table></td>
       <?php
               }
               /*   变量$i 自加 1，while 循环结束  */
               $i++;
          }
       ?>
   </tr>
```

通过上述代码可以发现，预览图片和显示图片实际调用的都是同一页面 image.php。image.php 页用来显示图片的实际效果，实现代码很简单，如下所示。

```
<?php
     include "Conn/conn.php";                                                          //包含数据库连接文件
    $query="select id,tpmc,file,author,scsj from tb_tpsc where id=".$_GET['recid'];   //根据 ID 生成查询语句
    $result=mysql_query($query);                                                       //执行查询语句
    if(!$result) die("error: mysql query");                                            //判断是否有返回结果
    $num=mysql_num_rows($result);                                                      //计算结果集个数
    if($num<1) die("error: no this recorder");
    $data = mysql_result($result,0,"file");                                            //返回图片数据
     echo $data;                                                                       //输出图片
?>
```

25.8.5　删除图片的实现过程

视频讲解：光盘\TM\Video\25\删除图片的实现过程.exe

在博客中，不能保证每个用户所发表的文章或照片都是合法的，为了保证博客能够正常运营、保持良好的形象，管理员需要及时将一些非法内容进行删除。本博客中用于删除图片的关键代码如下：

```
<?php
     session_start();                                                  //开启 Session 支持
     include "check_login.php";                                        //包含权限检查文件
     include "Conn/conn.php";                                          //包含数据库连接文件
    if($_session[fig] != "1"){                                         //判断是否为管理员
          echo "<script>alert('您没有删除权限  ');history.go(-1);</script>";
     }
     $sql="delete from tb_tpsc where id=".$_GET['pic_id'];             //生成删除语句
    $result=mysql_query($sql);                                         //执行删除语句
     if($result){                                                      //根据$result 返回结果
          echo "<script>alert('图片删除成功!');location='browse_pic.php';</script>";
     }
     else{
```

```
            echo "<script>alert('图片删除操作失败!');history.go(-1);</script>";
        }
?>
```

上述代码中，首先使用$_GET[]全局数组接收通过 GET 方法传递的图片的 ID 值，然后使用 mysql_query()函数执行查询，根据该 ID 值删除指定的图片。

25.9　朋友圈模块设计

在博客中除了可以发表文章和上传图片或照片外，还可以结交一些网络朋友，本博客的朋友圈模块即可为用户提供相互交流的平台。

25.9.1　朋友圈模块概述

通过本博客的朋友圈模块，用户可以对好友信息进行添加、查询和删除等操作。为了保证用户隐私不会泄露，用户所添加的好友信息只有本人可以看到，其他用户即使管理员也不能随意查看。如果博客中出现恶意用户，并且被管理员删除，那么该用户的朋友圈也一并被删除。朋友圈模块框架图如图 25.26 所示。

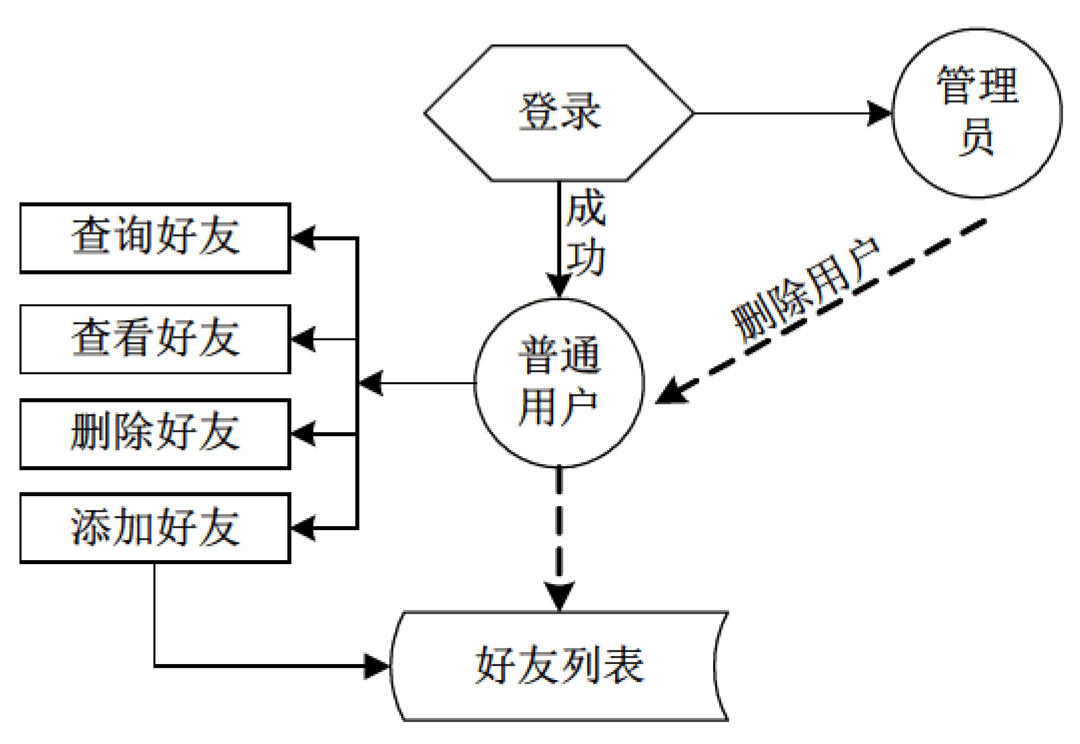


图 25.26　朋友圈模块框架图

25.9.2　朋友圈模块技术分析

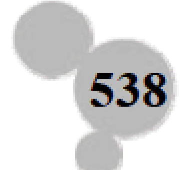
视频讲解：光盘\TM\Video\25\朋友圈模块技术分析.exe

为了有效查询朋友圈中的好友信息，该模块中使用模糊查询的方式对好友信息进行检索。SQL 语言中使用关键字 like 实现数据库中信息的模糊查询，一般采用如下两种方式对数据库中的信息进行模糊查询。

1．使用通配符“%”的 where 子句

通配符“%”表示 0 个或多个、任意长度和类型的字符，包括汉字。

示例 1：表示查找所有内容中包含“好”字的文章。



```
select * from tb_file where content like '%好%';
```

示例 2：查找所有包含“好”字或“高”字的文章，这时可以配合 or 运算符来使用，代码如下。

```
select * from tb_file where content like '%好%' or content like '%高%';
```

2．使用通配符“_”的 where 子句

通配符“_”表示匹配任意的单个字符。

示例 1：查找用户名只包含 5 个字符，其中后 4 个字符为 soft 的用户，代码如下。

```
select * from tb_user where regname like '_soft';
```

示例 2：查找所有以 t 开头，并且以 t 结尾的、中间包含 3 个字符的用户，代码如下。

```
select * from tb_user where regname like 't___t';
```

查找的结果为：tsoft。

注意：使用 MySQL 进行模糊查询要注意编码问题。如果编码不统一，那么查询时就容易查不到数据，或返回的数据不匹配。所以在安装 MySQL 时，要保持和系统编码的统一。常用的编码格式有 GB2312、ISO-8859-1、UTF-8 和 GBK 等。

25.9.3　查询好友的实现过程

视频讲解：光盘\TM\Video\25\查询好友的实现过程.exe

单击“朋友圈管理”/“查询朋友信息”超链接即可进入查找好友圈模块，如图 25.27 所示。查找好友时首先在“查询条件”下拉列表框中选择按什么条件进行查询，然后在“关键字”文本框中输入要查询的关键字，最后单击“检索”按钮即可查找出相关的好友信息。



图 25.27　查询好友页面的运行结果

查询页面所涉及的表单及表单元素说明如表 25.14 所示。

表 25.14　查询好友页面的表单及表单元素

元 素 类 型	元 素 名 称	属 性 设 置	含　　义
form	myform	method="post" action="query_friend.php" onSubmit= "return check();"	查询好友表单
select	sel_tj	<option value="name" selected>姓名</option> <option value="id">编号</option>	查询条件选择
text	sel_key	id="sel_key" size="30"	查询关键字
submit	submit	type="submit" name="submit" value="检索"	提交按钮

查询处理页通过 POST 方法接收到查询条件及查询关键字后，根据提交关键字模糊查询语句，执行 SQL 语句并返回查询结果。如果没有输入关键字，则弹出提示框；如果没有查找到任何结果，则输出“Sorry!没有您要找的朋友!”。查询处理页的关键代码如下：

```
<?php
session_start();                                    //开启 Session 支持
include "Conn/conn.php";                            //包含数据库连接文件
include "check_login.php";                          //包含权限检查文件
?>
<script language="javascript">
/*  JavaScript 脚本，验证输入信息，如果为空，则弹出提示框  */
function check(form){
    if (document.myform.sel_key.value==""){
        alert("请输入查询条件!");
        myform.sel_key.focus();
        return false;
    }
}
</script>
…
<?php
    if ($_POST[sel_key]!=""){
        $tj=$_POST[sel_tj];                         //接收查询字段
        $key=$_POST[sel_key];                       //接收查询关键字
        $sql=mysql_query("select * from tb_friend where $tj like '%$key%' and username='$_SESSION[username]'");
        $result=mysql_fetch_array($sql);            //执行查询语句
        if($result==false){
            echo ("[<font color=red>Sorry!没有您要找的朋友!</font>]");
        }else{
?>
<!--  显示记录表单  -->
<?php
        }
    }
?>
```

25.10　项　目　发　布

将网站发布到 Internet 的方法有多种，最常用的方式有使用独立服务器发布站点和租用虚拟主机空间发布站点。

25.10.1　通过个人服务器发布项目

视频讲解：光盘\TM\Video\25\通过个人服务器发布项目.exe

完成网站的开发后，还需要将网站发布到 Internet 中，如果使用独立 IP 并且拥有自己的服务器，那么发布网站相对简便，只需安装 PHP 运行环境并填写独立 IP 地址，并将申请的域名解析到服务器 IP 即可。如果希望在一台服务器上运行多个站点，那么可以配置 Apache 服务器的 httpd.conf 文件中 Virtual Hosts 模块。

例如，在独立 IP 为 221.8.65.**的服务器中发布网站 www.mingri**.com 和 www.mr**.com。

（1）正确填写服务器 IP 地址、域名地址等信息。

（2）将 www.mingri**.com 和 www.mr**.com 都解析到 221.8.65.**这个地址。

（3）配置 httpd.conf 文件（假设 www.mingri**.com 域名所对应的站点文件存放在服务器 c 盘的 mingri**目录下，www.mr**.com 存放在 c 盘的 mr**下）。

```
<VirtualHost *:80>
    DocumentRoot c:/mingri**
    ServerName www.mingri**.com
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot c:/mr**
    ServerName www.mr**.com
</VirtualHost>
```

（4）将项目文件分别复制到指定的目录中。

（5）重新启动 Apache 服务器生效。

25.10.2　通过租用空间发布项目

视频讲解：光盘\TM\Video\25\通过租用空间发布项目.exe

要使用租用虚拟主机空间的方式发布站点，需要经过注册域名、申请空间、解析域名和上传网站 4 个步骤。首先介绍一下注册域名。

1．域名申请

域名就是用来代替 IP 地址，以方便记忆及访问网站的名称。例如，www.mingribook.com 就是明日图书网的域名，www.yahoo.com.cn 就是中文雅虎的域名。域名需要到指定的网站中注册购买，如

www.net.com（万网）和 www.xinnet.com（新网）等。

购买注册域名的步骤如下：

（1）登录域名服务商网站。

（2）注册会员。如果不是会员则无法购买域名。

（3）进入域名查询页面，查询要注册的域名是否已经被注册。

（4）如果用户欲注册的域名未被注册，则进入域名注册页面，在该页面中填写相关的个人资料。

（5）填写成功后，单击“购买”按钮，注册成功。

（6）付款后，等待域名开启。

2．申请空间

域名注册完毕后，下一步是申请空间。空间可以使用虚拟主机或租借服务器。目前，许多企业建立网站都采用虚拟主机，这样既节省了购买机器和租用专线的费用，同时也不必聘用专门的管理人员来维护服务器。申请空间的步骤如下：

（1）登录虚拟空间服务商网站。

（2）注册会员（如果已有会员账号，则直接登录即可）。

（3）选择虚拟空间类型（空间支持的语言、数据库、空间大小、流量限制等）。

（4）确定机型后，直接购买。

（5）进入到缴费页面，选择缴费方式。

（6）付费后，空间在 24 小时内开通，随后即可使用此空间。

3．将域名解析到服务器

域名和空间购买成功后，接下来需要将域名地址指向虚拟服务器的 IP。进入域名管理页面，添加主机记录。一般先要输入主机名，注意不包括域名。例如，解析 www.bccd***.com，如果要添加该主机，只需输入 www 即可，后面的 bccd***.com 不需要填写。接下来填写 IP 地址。最后单击“确定”按钮，剩下的工作交给域名服务器即可。如果想添加多个主机名，则重复上面的操作即可。

4．上传网站

最后就是上传网站程序。上传网站需要使用 FTP 软件。如果使用 Dreamweaver，则可以直接在 Dreamweaver 中上传。这里以 CuteFTP 为例讲解上传步骤。

（1）打开 FTP 软件。

（2）选择 File/Site-Manager 命令，弹出站点面板。

（3）单击 New 按钮，新建一个站点。

（4）在 Label for site 文本框中输入站点名。

（5）在 FTP Host Address 文本框中输入域名。

（6）在 FTP site User Name 文本框中输入用户名。

（7）在 FTP site Password 文本框中输入密码。

（8）单击 Edit 按钮，弹出编辑窗口。

（9）选中 Use PASV mode 和 Use firewall setting 复选框。

（10）单击“确定”按钮。

（11）单击 Connet 按钮连接到服务器。



（12）连接服务器后，在左侧的本地页面中右击需要上传的文件，单击“上传”按钮即可。

（13）如果上传过程中出现错误，选择“继续上传”命令即可。

（14）上传成功后，关闭 FTP 软件。

25.11　开发常见问题与解决

视频讲解：光盘\TM\Video\25\开发常见问题与解决.exe

在项目开发过程中，免不了会遇到一些问题，其实问题并不可怕，关键是需要开发人员能够认真细心地去面对、去解决。下面将介绍如何处理和解决博客开发过程中常见的问题。

1．数据库乱码问题

首先查看本博客的数据库连接文件，代码如下：

```
<?php
$link=mysql_connect("localhost","root","root");         //建立与 MySQL 数据库的连接
mysql_select_db("db_blog",$link);                        //选择数据库
mysql_query("set names gb2312");                         //设置字符集
?>
```

最初在开发本博客时，并没有加上“mysql_query("set names gb2312")”这句代码，可是在调试程序时，发现从数据库提取的所有中文字符都为乱码，如图 25.28 所示。

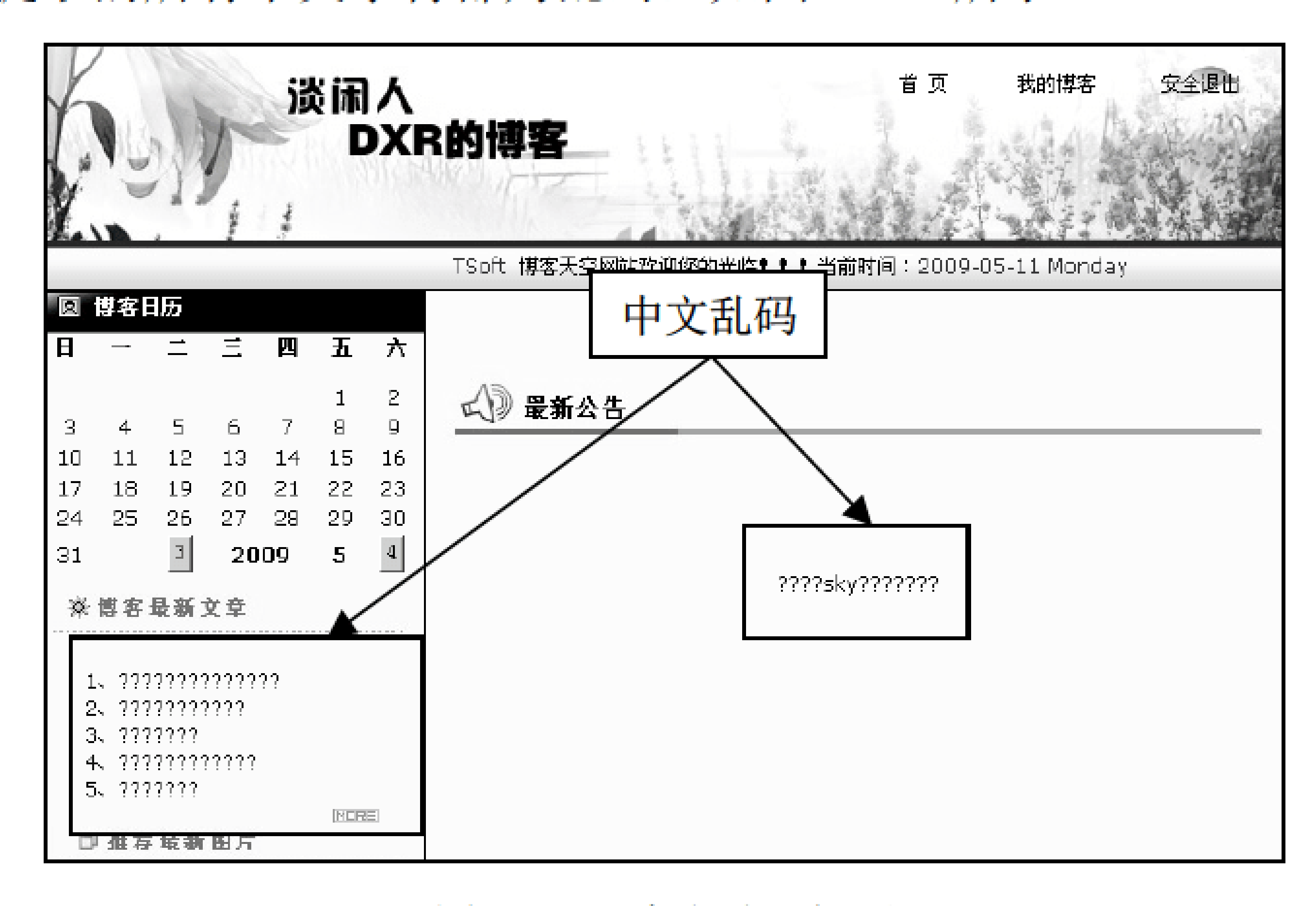


图 25.28　中文乱码问题

这是没有正确设置数据库中的字符集所导致的乱码错误。“mysql_query("set names gb2312")”这句代码的作用就是设置 MySQL 数据库的字符集为 GB2312，当然也可以使用这句代码设置 MySQL 数据库的字符集为其他类型，如 UTF-8 等。

2．接收 GET 方法或 POST 方法提交数据问题

在接收 GET 方法或 POST 方法提交的数据时，一般使用$_GET 或$_POST 全局数组进行接收。

在实际开发过程中，有时不用这两个全局数组进行接收，而直接用与传递的参数同名或与提交表

第26章

应用 Smarty 模板开发电子商务平台

（视频讲解：281 分钟）

电子商务（Electronic Commerce），简称EC，是指整个事务活动和贸易活动的电子化，它通过先进的信息网络，将事务活动和贸易活动中发生关系的各方有机地联系起来。电子商务网站实际上就是企业为消费者提供的网上虚拟购物商城，在该网站中，用户可以购买任何商品，而管理员可以对商品和订单等信息进行管理。通过本章的学习，读者不仅可以丰富自己的基础知识、掌握大量网站制作技巧及PHP常用函数的使用方法，更能掌握网络编程的设计思路、方法和过程，快速提高PHP开发能力和设计水平。

通过阅读本章，可以学习到：

- **如何进行系统分析**
- **如何编写项目计划书**
- **数据库设计流程**
- **搭建系统架构**
- **简单的树状菜单**
- **如何灵活运用 JS 和 Ajax 技术**
- **如何灵活运用购物车**
- **如何灵活运用 Smarty 模板开发程序**
- **ADODB 在实际中的应用**

26.1 项 目 描 述

开发电子商务平台实质上就是建立一个虚拟的购物超市，当在超市选购商品时，首先应该将商品放到购物车中，待挑选好所有商品之后就可以到收银台去结算，根据结算单据付款。开发电子商务平台的原理与在超市购物的原理是相同的：客户通过网页浏览商品，选购自己喜欢的商品，将商品放入购物车中，当然也可以改变购买商品的数量或者清空购物车中的商品，在选购好商品后到收银台去结算，生成订单，填写并提交收货人信息，最终按照订单中提交的付款方式进行付款，等待送货上门，期间还可以登录平台查看自己的订单。

企业就是这个虚拟超市的管理者，可以对登录的用户、商品信息、商品类别、订单和公告等内容进行管理。

本系统建立一个电子商务平台，实现网上购物的功能，包括从客户最初的商品浏览到最后的购物结算、订单查看，以及企业的商品信息、商品类别、客户信息和订单的管理。

系统开发应用 PHP 语言+MySQL 数据库，通过 Smarty 模板实现网页动静分离，通过 ADODB 操作数据库，其中还应用到 Ajax、Fram 框架、JavaScript、CSS 和 DIV 等技术。

系统中典型的功能模块是：找回密码、Ajax 无刷新注册验证、GD2 图像验证码、购物车、收银台、树状菜单和商品信息管理。

26.2 系 统 分 析

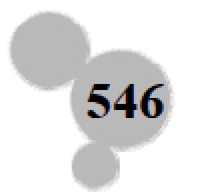
26.2.1 需求分析

中国的市场经济日趋成熟，中国企业面对的竞争压力也越来越大，企业正在寻求一种高效、便于客户购物和支付的购物形式，而电子商务这种新的商业运营模式完全符合商家们的要求，同时也可以满足消费者的需求。消费者能够在网上实现互动的交流及足不出户地购买产品，向企业发表自己的意见、服务需求及有关投诉，并且可以通过网站的交互式操作向企业进行产品的咨询，以此来获取相应的回馈信息及技术支持。

电子商务被越来越多的商家和消费者认同，必将成为 21 世纪网络发展的主流，而网上购物将成为一种购物时尚。目前国内企业正纷纷加入到阿里巴巴等一系列大型电子商务网站中，通过网络进行 B2B（企业对企业）的网上交易之后，也充分认识到网络交易将成为未来商品交易的重要组成部分，在通过网络宣传企业的同时，也逐步扩大企业自身的网络销售渠道，建立起自己的电子商务平台，完成了从 B2B 到 B2C（企业对个人）进行网络交易的过渡，大大提高了企业的经济效益。因此，企业开发属于自己的电子商务平台已经势在必行。

26.2.2 编写项目计划书

项目计划书是项目负责人写给投资者的一份详细的开发报告，包括开发背景、开发原因、可行性



电子商务平台项目计划书

一、项目开发背景

自 20 世纪 90 年代，互联网的蓬勃发展，为企业提供了一个全新的机遇。企业网站、电子商务成为热门话题。其中，电子商务更是关系到经济结构、产业升级和国家整体经济竞争力。为此，我国已经将发展电子商务列为信息化建设的重要内容，并努力创造条件，积极地推进电子商务的发展。

据美国在线（AOL）和 Henley Centre 联合进行的一项调查显示：国外有 80%的受调查者会选择网上购物或寻求帮助，10%的受调查者会选择熟悉的品牌或厂商来购买。而在国内，自 1997 年拉开了电子商务的序幕，短短的 10 年时间里，全国已有 4 万家商业网站，几乎每天都有新的网站诞生，厂商所在地也从上海、广州、深圳等沿海发达地区扩展到全国各大中城市。

二、电子商务的概念及特点

电子商务（Electronic Commerce），简称 EC，是目前发展较快的一种商务模式。不同领域的人群对 EC 的理解各有不同，综合来说，电子商务是通过互联网进行的商务活动，它已经渗透了各个领域，如服务、金融、销售等。

目前来说，电子商务主要有两种模式。一种是纯粹的电子商务网站，如阿里巴巴，它为交易双方提供了一个丰富的信息平台；另一种是传统产业的电子商务模式，它借助互联网进行特定商品的销售及采购。

三、电子商务的优势

电子商务的优势是显而易见的，主要有以下几个方面：

- 大大节约了企业的成本预算，如人工费、场地费等。
- 降低了产品的零售价格。因为是和消费者直接交流沟通，取消了代理、批发、商场等费用，在同行业的竞争中占据优势。
- 扩展了业务范围，只要能够上网的人，就有机会成为客户，一定程度上消除了地域的影响。
- 加速了资本流通。
- 实现 24 小时连续服务。

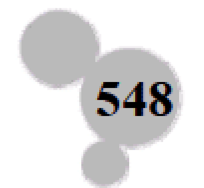
四、可行性分析

- 市场前景。数据表明：2007 年中国 B2B 电子商务交易规模为 12500 亿元，比 2006 年增长 25.5%。预计未来两年我国 B2B 电子商务交易规模将继续高速增长，2008 年将达到约 16200 亿元，2009 年交易规模有望达 21300 亿元。B2C 网站总收入为 52.2 亿元，2009 年有望达到 98.6 亿元。而且，随着网络购物环境的好转，未来两年 B2C 电子商务交易模式将更受欢迎，用户数和年平均消费金额均会提高。这里所蕴含的商机毋庸置疑。
- 技术可行性。电子商务相关的技术（如认证加密技术、数据备份技术、网上交易技术）已经十分完善，能够保障系统正常、无错地运行。
- 经济可行性。通过网络进行商务活动，可以足不出户地完成交易。既可节省交通费，又减少了务工人员费用，同时，前期可以使用小型服务器或虚拟空间以减少成本预算。

五、项目实施的技术方案

- 为了节约成本，使用 PHP 做开发，数据库使用 MySQL。
- 采用 Smarty 模板，可以最大限度地加快网页访问速度。
- 考虑到以后可能升级数据库，使用 ADODB 类库连接。
- 使用 Ajax 技术来提高交互性。
- 提供多种付款方式，包括网银、邮购、汇款等。
- 提供在线咨询。

第 2 页共 4 页



● 操作简单容易，只需要简单指点便可使用，无须专业人员。

六、总体投入

报价

顶级域名	.com 或.net 结尾（英文域名）	￥100/年
虚拟主机	智尊 U300（250MB/赠送 125MB 邮局/50MB mysql/php/perl）	￥750/年
电子商务平台	电子商务网上交易系统	￥20000/套（包括一年的免费维护，如需二次开发费用另算）
留言板	用户提交留言，不需审核立即显示，管理员可随时回复删除相关主题，是一套人性化的沟通工具，有助于搞好客主关系	￥500/个
友情链接	管理员可随时添加相关链接	￥500/个
公告版		￥500/个
LOGO		￥200/个
搜索引擎优化	使网站能够被搜索引擎抓取，并排名靠前，利于网站的推广	赠送

共计：域名+虚拟主机+电子商务系统+留言板+友情链接+公告版=¥22550，以后每年只需交¥850（域名+主机）即可。

七、人员安排

人员安排

项目负责人	1 名（邹甜甜）	负责项目的总体设计以及与企业沟通
系统策划	1 名（刘明松）	负责网站的策划和制作
网站开发人员	4 名（潘明远、孙海、刘忻南、梁焱）	负责系统后台开发
美工	1 名（董朋）	负责页面和 LOGO 设计
测试人员	1 名（刘杏）	负责系统的后期调试

八、后期维护

在系统正式使用的一年时间内，将对系统进行免费维护（添加系统功能或修改框架除外），及时修改程序缺陷，还为客户提供技术培训。同时，保证运行速度和系统安全。

九、系统功能模块

本系统分前台和后台两部分。

● 前台包括

■ 登录及注册模块：包括会员的注册、登录和找回密码功能。

■ 商品模块：包括商品的分类浏览和查看。

■ 商品搜索模块：包括精确查询和模糊查询两种模式。实现对商品任意信息的搜索。

■ 购物车模块：帮助会员完成购物功能。

■ 在线支付模块：支持工行、招行等各个银行的信用卡、支付卡。

■ 公告栏模块：最新的咨询、活动、声明等。

■ 友情链接模块：行业内网站之间的互换链接栏目的要求，实现网站的免费推广与宣传。

■ 留言模块：包括留言板、在线 QQ 等客户和企业的交流平台。

● 后台包括

■ 商品管理：包括对商品的增删改查等操作和商品的类别管理。

■ 会员管理模块：包括对会员的信息和留言管理。

■ 订单管理模块：管理员可以对客户提交的订单进行编辑、查看和处理。

26.3　系 统 设 计

26.3.1　系统目标

根据电子商务平台的操作流程，结合客户提供的需求，以及对实际情况的考察与分析，将电子商务平台的特点总结如下：

☑ 页面设计简洁、明快，突出重点，以吸引用户目光为目的。
☑ 前台操作流程清晰，确立从商品浏览、购物车、提交订单、查看订单的基本流程。
☑ 后台管理目标明确，包括对用户、商品信息、订单和公告等信息的管理。
☑ 可操作性高，避免复杂的、有异议的链接。
☑ 浏览速度快，尽量避免长时间打不开页面的情况发生。
☑ 商品浏览以实物图片为参照，图像清晰，文字醒目。
☑ 商品查询功能，可以通过商品的各个属性来搜索。
☑ 商品分类功能，为用户购物提供导向。
☑ 网站架构合理，数据库稳定。
☑ 易维护，并提供二次开发支持。

26.3.2　系统功能结构

电子商务平台分前台系统和后台系统。下面分别给出前、后台的系统功能结构图。电子商务前台系统功能结构图如图 26.1 所示。

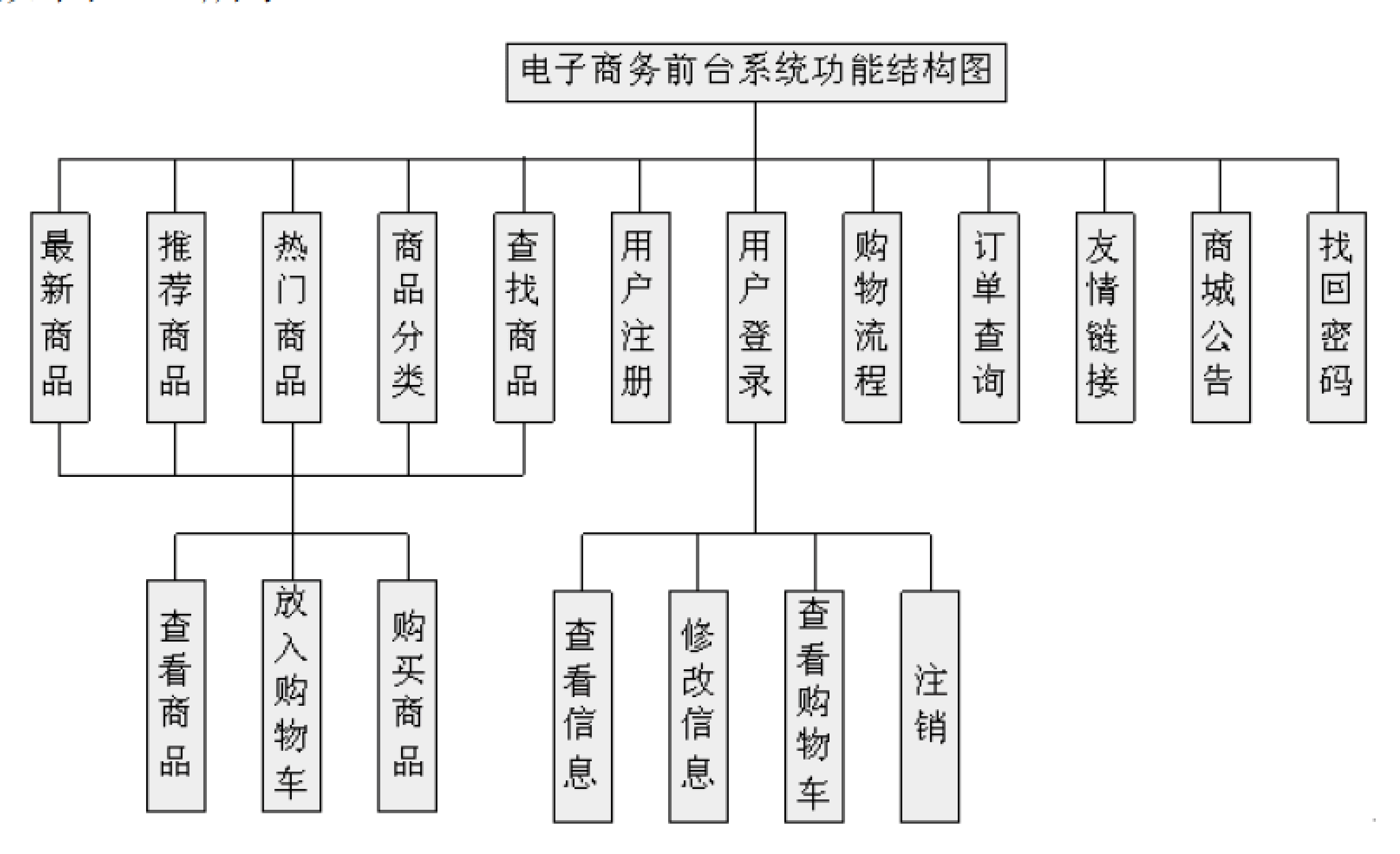


图 26.1　电子商务前台系统功能结构图

电子商务后台系统功能结构图如图 26.2 所示。

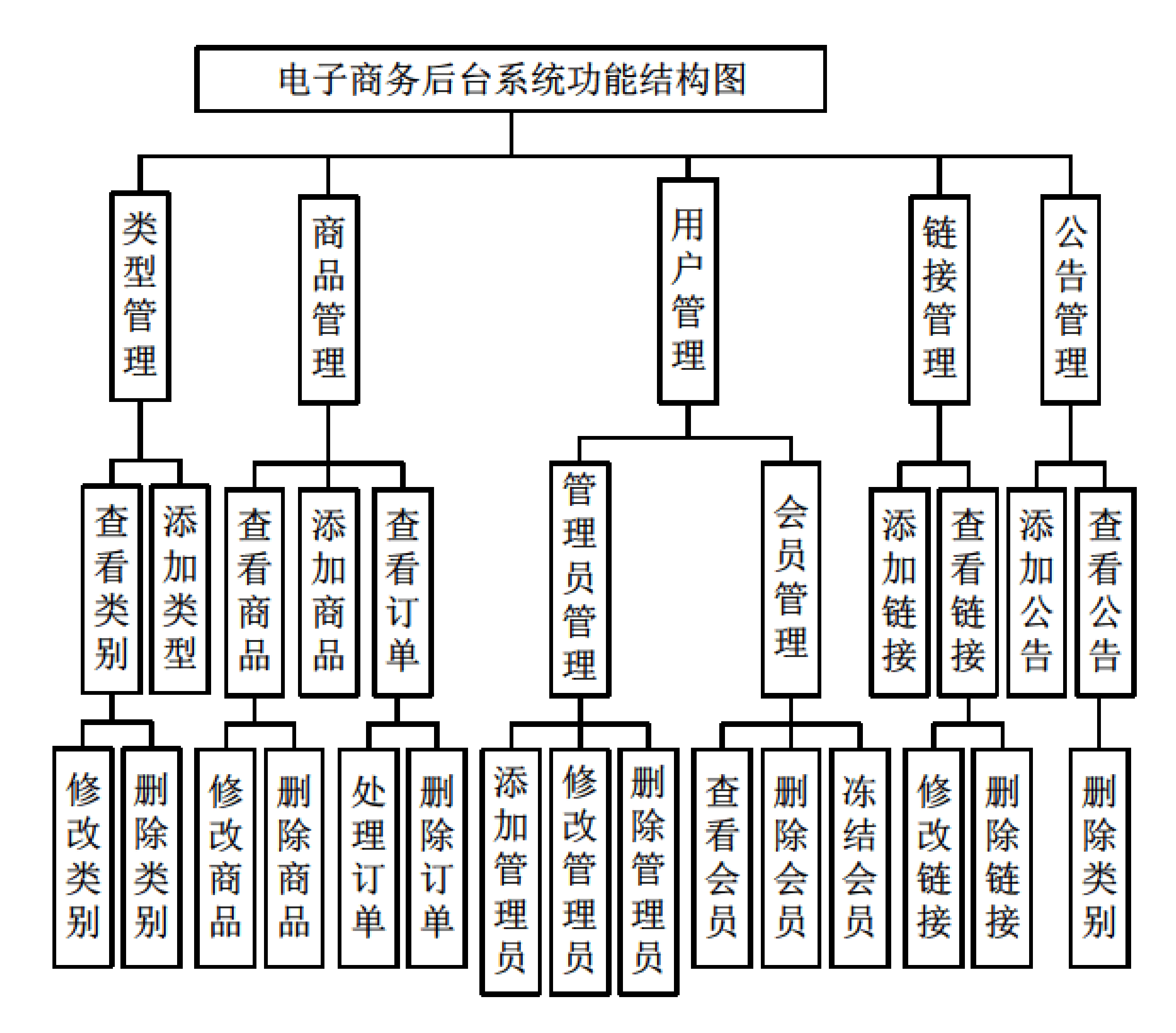


图 26.2　电子商务后台系统功能结构图

26.3.3　系统流程图

为了便于开发人员了解系统各功能模块之间的联系及完整的购物流程，下面给出系统的购物流程图，如图 26.3 所示。

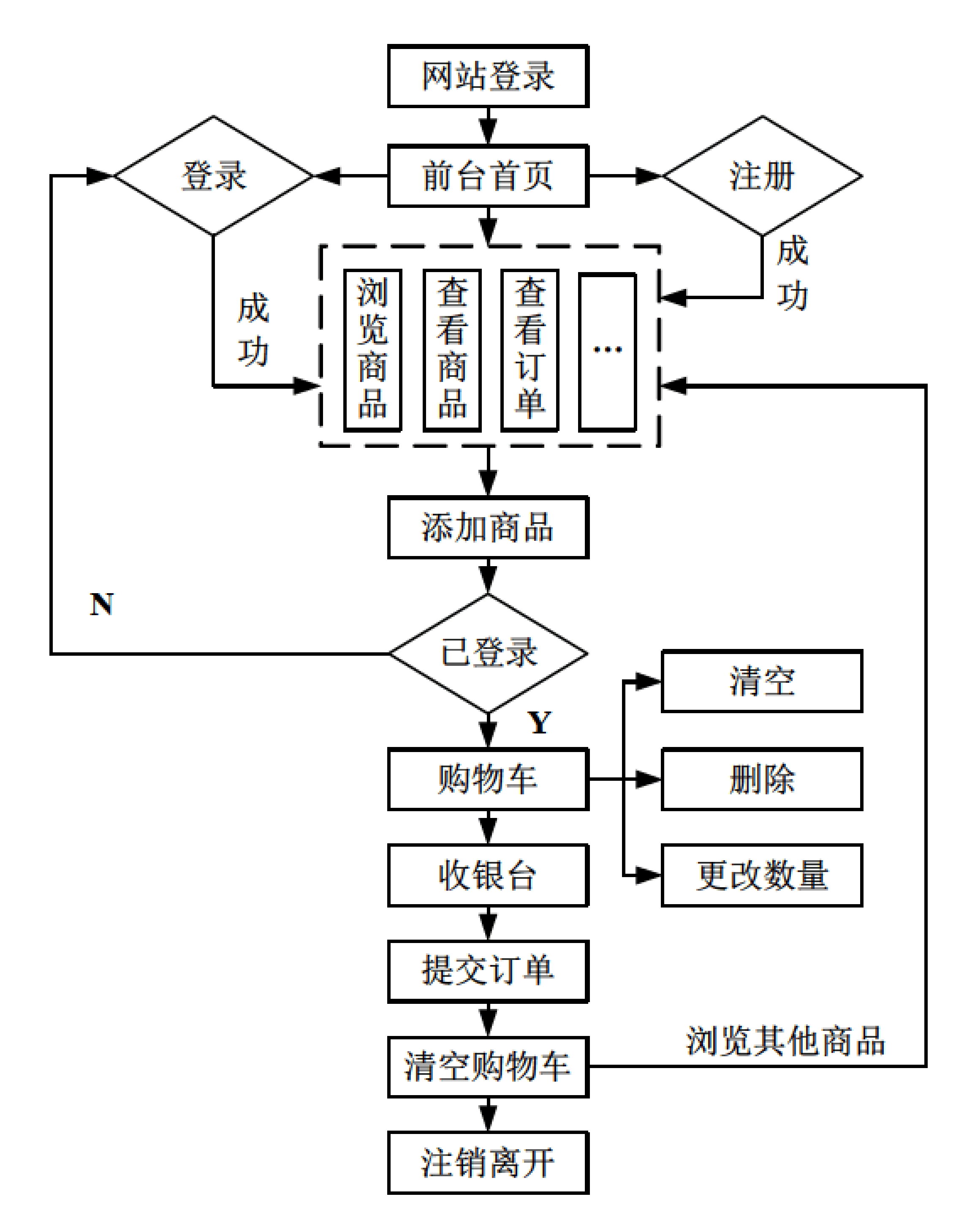
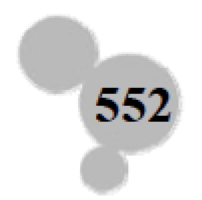


图 26.3　系统流程图



26.4　系统开发环境

在开发电子商务平台时，该项目应用到的软件如下。

☑ 服务器端

- 操作系统：Windows 2003 Server/Linux（推荐）。
- 服务器：Apache 2.2.3。
- PHP 软件：PHP 5.1.6。
- 数据库：MySQL 5.0.24。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- ADODB：ADODB 5.0。
- Smarty 模板：Smarty-2.6.19。
- 开发工具：Dreamweaver 8。
- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

☑ 客户端

- 浏览器：推荐 IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

26.5　数据库与数据表的设计

视频讲解：光盘\TM\Video\26\数据库与数据表的设计.exe

无论是什么系统软件，其最根本的功能就是对数据的操作与使用。所以，一定要先做好数据的分析、设计与实现，才能更好地完成对应的功能模块。

26.5.1　数据库分析

根据需求分析和系统的功能结构图，找出需要保存的信息数据（也可以理解为现实世界中的实体），并将其转化为原始数据（属性类型）形式。这种描述现实世界的概念模型，可以使用 E-R 图来表示，也就是实体-联系图。最后将 E-R 图转换为关系数据库。这里重点介绍几个 E-R 图。

1．会员信息实体

会员信息实体包括编号、用户名、密码、E-mail、身份证号、联系电话、QQ 号、密码提示、密码答案、邮编、注册时间、真实姓名等属性。会员信息实体 E-R 图如图 26.4 所示。

2．商品信息实体

商品信息实体包括编号、名称、上市时间、添加时间、型号、图片、库存、销售、商品类型、会员价、市场价、是否打折等属性。商品信息实体 E-R 图如图 26.5 所示。

实体的 E-R 图如图 26.6 所示。

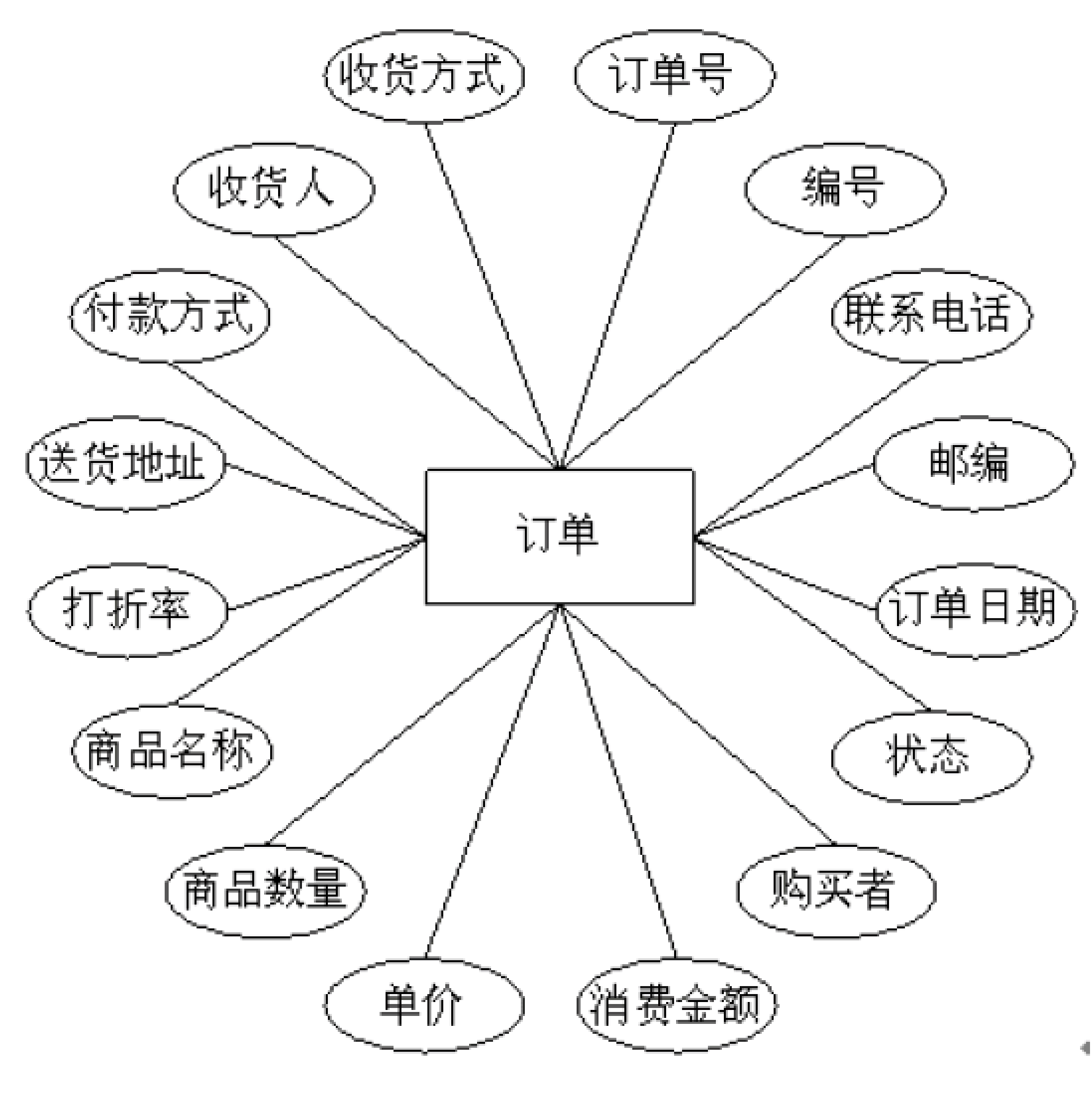


图 26.6　商品订单 E-R 图

除了上面介绍的 3 个 E-R 图，还有公告实体、管理员实体、类型实体、友情链接实体，限于篇幅，这里不再赘述。

26.5.2　创建数据库和数据表

系统 E-R 图设计完成后，接下来根据 E-R 图来创建数据库和数据表。关于数据库和数据表的创建可参考第 17 章 MySQL 数据库基本操作，这里不再赘述。首先看一下电子商务平台都使用了哪些数据表，如图 26.7 所示。

服务器: localhost ▸ 数据库: db_database26

表	类型	整理	说明
tb_admin	MyISAM	gb2312_chinese_ci	管理员信息表
tb_class	MyISAM	gb2312_chinese_ci	商品类型表
tb_commo	MyISAM	gb2312_chinese_ci	商品信息表
tb_form	MyISAM	gb2312_chinese_ci	商品订单表
tb_links	MyISAM	gb2312_chinese_ci	友情链接表
tb_opinion	MyISAM	gb2312_chinese_ci	商品评论表
tb_public	MyISAM	gb2312_chinese_ci	公告信息表
tb_user	MyISAM	gb2312_chinese_ci	会员信息表

图 26.7　电子商务数据表

下面来看各个数据表的结构和字段说明。

☑　tb_admin（管理员信息表）

管理员信息表主要用于存储管理员的信息。该数据表结构如图 26.8 所示。

☑　tb_class（商品类型表）

商品类型列表主要用于添加商品的类别，可以设定多个子类别，目前最多只能到二级子类别。该数据表结构如图 26.9 所示。

服务器: localhost ▸ 数据库: db_database26 ▸ 表 : tb_admin

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	gb2312_chinese_ci		否			管理员账号
pwd	varchar(20)	gb2312_chinese_ci		否			管理员密码

图 26.8　管理员信息表结构

服务器: localhost ▸ 数据库: db_database26 ▸ 表 : tb_class

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(20)	gb2312_chinese_ci		否			类型名称
level	int(1)			否	1		等级

图 26.9　商品类型表结构

☑　tb_commo（商品信息表）

商品信息表主要用于存储关于商品的相关信息。该数据表结构如图 26.10 所示。

☑ tb_form（商品订单表）

商品订单表主要用于存储商品的订单信息。该数据表结构如图 26.11 所示。

服务器: localhost ▸ 数据库: db_database26 ▸ 表 : tb_commo

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	gb2312_chinese_ci		否			商品名称
pics	varchar(200)	gb2312_chinese_ci		否			商品图片
info	mediumtext	gb2312_chinese_ci		否			商品介绍
addtime	date			否			添加时间
area	varchar(50)	gb2312_chinese_ci		否			商品产地
model	varchar(50)	gb2312_chinese_ci		否			商品型号
class	varchar(50)	gb2312_chinese_ci		否			商品类别
brand	varchar(50)	gb2312_chinese_ci		否			商品品牌
stocks	int(4)			否	1		商品库存
sell	int(4)			否	0		商品销量
m_price	float			否			市场价格
v_price	float			否			会员价格
fold	int(2)			否	9		打折率
isnew	int(1)			否	1		是否新品
isnom	int(1)			否	0		是否推荐

图 26.10　商品信息表结构

服务器: localhost ▸ 数据库: db_database26 ▸ 表 : tb_form

字段	类型	整理	属性	Null	额外	说明
id	int(4)			否	auto_increment	自动编号
formid	varchar(125)	gb2312_chinese_ci		否		订单号
commo_id	varchar(100)	gb2312_chinese_ci		否		商品id
commo_name	varchar(50)	gb2312_chinese_ci		否		商品名称
commo_num	varchar(100)	gb2312_chinese_ci		否		商品数量
agoprice	varchar(50)	gb2312_chinese_ci		否		商品价格
fold	varchar(50)	gb2312_chinese_ci		否		商品折率
total	varchar(50)	gb2312_chinese_ci		否		总金额
vendee	varchar(50)	gb2312_chinese_ci		否		订单用户
taker	varchar(50)	gb2312_chinese_ci		否		收货人
address	varchar(200)	gb2312_chinese_ci		否		收货地址
tel	varchar(20)	gb2312_chinese_ci		否		移动电话
code	varchar(10)	gb2312_chinese_ci		否		邮编
pay_method	varchar(20)	gb2312_chinese_ci		否		付款方式
del_method	varchar(20)	gb2312_chinese_ci		否		送货方式
formtime	timestamp			否		订单时间
state	int(1)			否		订单状态

图 26.11　商品订单表结构

☑ tb_public（商品公告表）

商品公告表主要用于展示网站的最新活动和最新消息，包括公告时间、公告标题和公告内容。该数据表结构如图 26.12 所示。

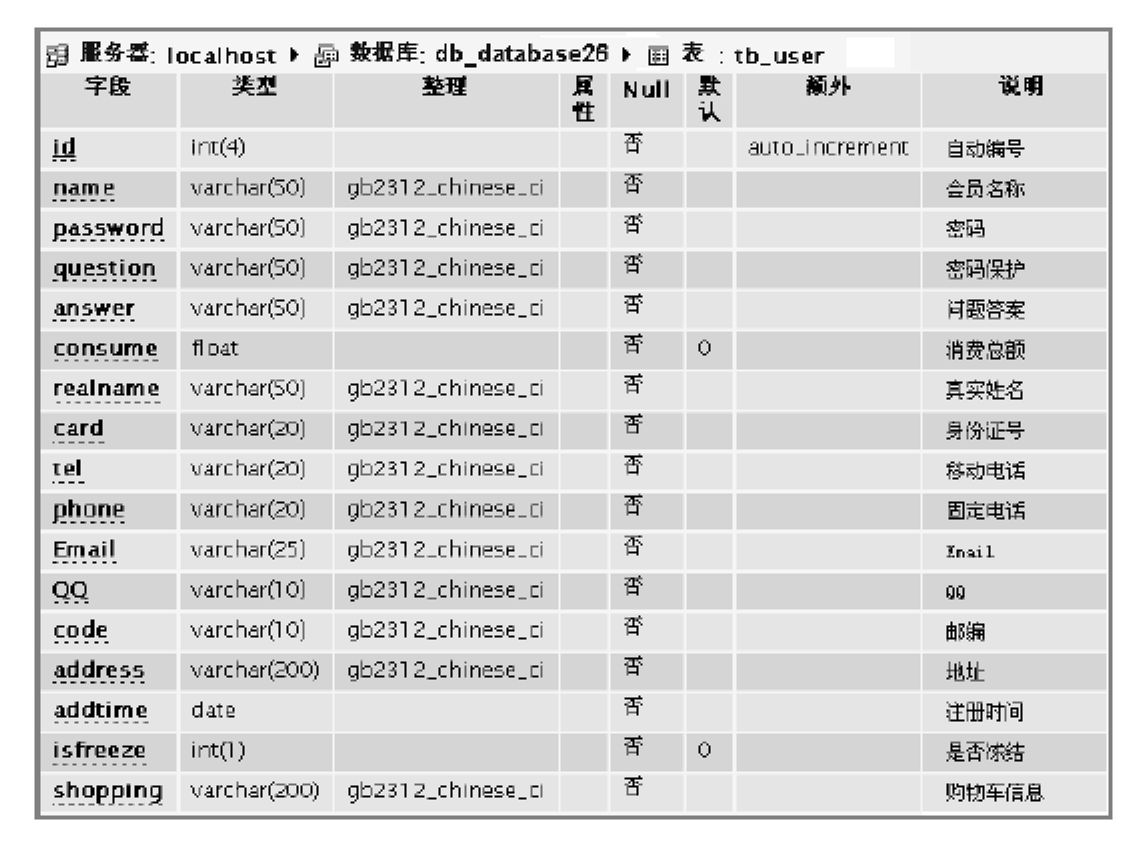
☑ tb_user（用户信息表）

用户信息表主要用于存储用户的基础信息。该数据表结构如图 26.13 所示。

服务器: localhost ▸ 数据库: db_database26 ▸ 表 : tb_public

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
title	varchar(50)	gb2312_chinese_ci		否			公告标题
content	mediumtext	gb2312_chinese_ci		否			公告内容
is_time	date			否			发布时间

图 26.12　商品公告表结构



服务器: localhost ▸ 数据库: db_database26 ▸ 表 : tb_user

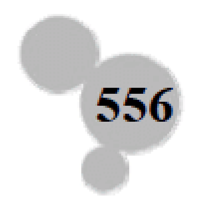
字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	gb2312_chinese_ci		否			会员名称
password	varchar(50)	gb2312_chinese_ci		否			密码
question	varchar(50)	gb2312_chinese_ci		否			密码保护
answer	varchar(50)	gb2312_chinese_ci		否			问题答案
consume	float			否	0		消费总额
realname	varchar(50)	gb2312_chinese_ci		否			真实姓名
card	varchar(20)	gb2312_chinese_ci		否			身份证号
tel	varchar(20)	gb2312_chinese_ci		否			移动电话
phone	varchar(20)	gb2312_chinese_ci		否			固定电话
Email	varchar(25)	gb2312_chinese_ci		否			Email
QQ	varchar(10)	gb2312_chinese_ci		否			QQ
code	varchar(10)	gb2312_chinese_ci		否			邮编
address	varchar(200)	gb2312_chinese_ci		否			地址
addtime	date			否			注册时间
isfreeze	int(1)			否	0		是否冻结
shopping	varchar(200)	gb2312_chinese_ci		否			购物车信息

图 26.13　用户信息表结构

此外还有友情链接表和商品评论表，限于篇幅，这里不再介绍，读者可参考本书配套光盘中的数据库文件。

26.6　搭建系统框架

编写代码之前，可以把系统中可能用到的文件夹先创建出来（如创建一个名为 images 的文件夹，



用于保存程序中所使用的图片），这样不但可以方便以后的开发工作，也可以规范系统的整体架构。因为本项目使用的是 Smarty+ADODB 技术，所以目录较多。下面来看本系统的目录结构（到三级目录），如图 26.14 所示。

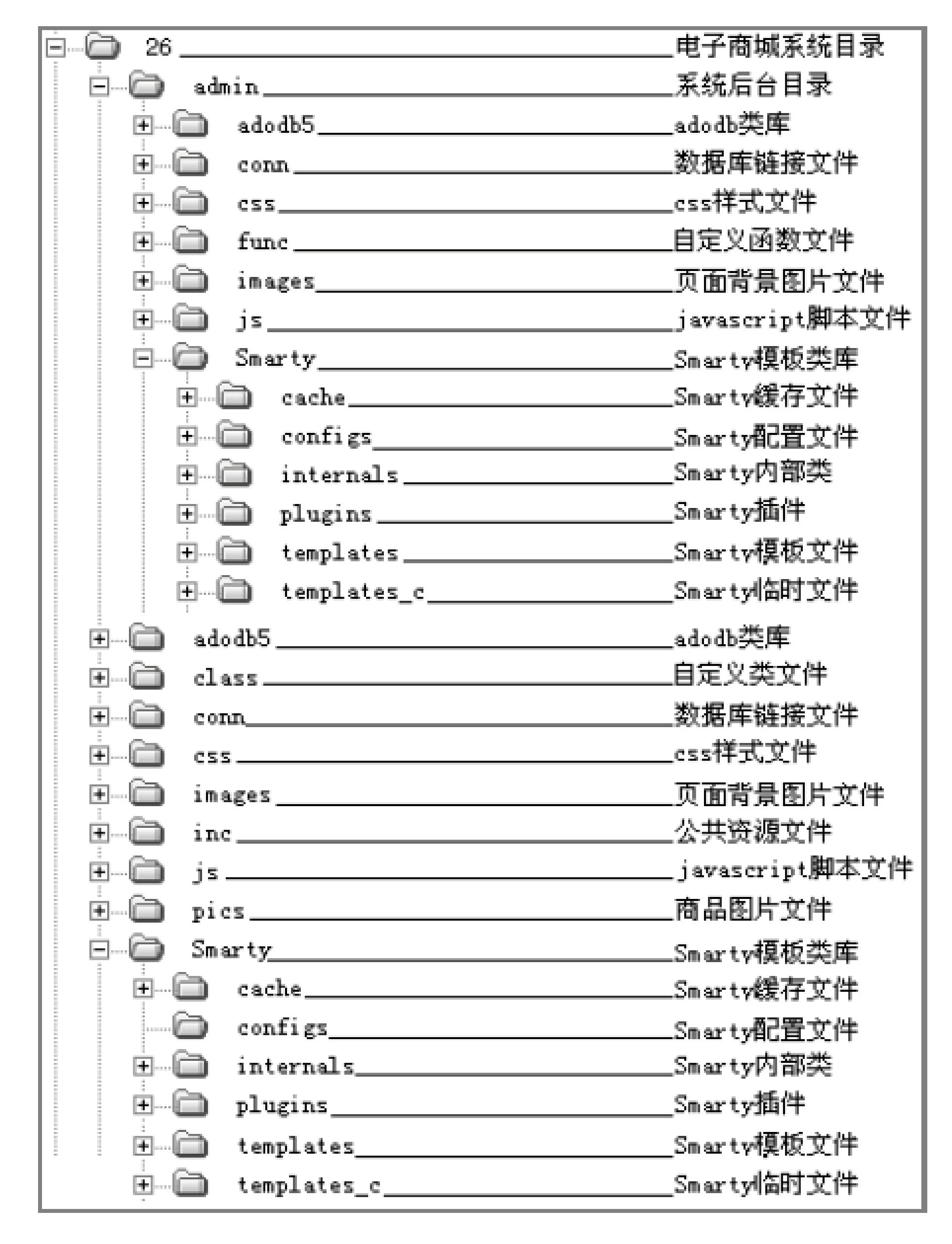


图 26.14　系统目录结构图

26.7　公共文件设计

视频讲解：光盘\TM\Video\26\公共文件设计.exe

公共文件就是将多个页面都可能使用到的代码写成单独的文件，在使用时只要用 include 或 require 将文件包含进来即可。如本系统中的数据库连接文件、Smarty 配置文件、CSS 样式文件、JS 脚本文件等。以前台系统为例，下面给出主要的公共文件。后台的公共文件与前台大同小异。

26.7.1　数据库连接文件

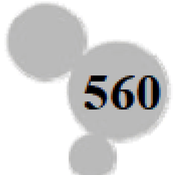
数据库连接文件，就是在读取或存储数据时连接数据库所使用到的函数。conn.php 文件代码如下。

```
<?php
    include_once ("adodb5/adodb.inc.php");                                    //载入 ADODB 类库
    $conn = ADONewConnection('mysql');                                        //连接 MySQL
    $conn->PConnect('localhost','root','root','db_database26') or die('connection error');   //db_database26 库
    $conn->Execute('set names gb2312');                                       //设置编码
    $ADODB_FETCH_MODE = ADODB_FETCH_BOTH;                                     //设置结果集存储方式
?>
```

```
}
.lk:visited {
	text-decoration: none;
	color: #350407;
}
.lk:hover {
	text-decoration: none;
}
.lk:active {
	text-decoration: none;
	color: #CCCCFF;
}
<!--   表格首行样式   -->
.first {
	background-color: #FFC4C4;
	border-bottom-width: 1px;
	border-bottom-style: solid;
	border-bottom-color: #FFC4C4;
	color: #000000;
	font-size: 14px;
	font-weight: bold;
}
<!--   左侧表格样式   -->
.left {
	border-left-width: 1px;
	border-right-width: 1px;
	border-bottom-width: 1px;
	border-left-style: solid;
	border-right-style: solid;
	border-bottom-style: solid;
	border-left-color: #FFC4C4;
	border-right-color: #FFC4C4;
	border-bottom-color: #FFC4C4;
}
<!--   中间表格样式   -->
.center {
	border-right-width: 1px;
	border-bottom-width: 1px;
	border-right-style: solid;
	border-bottom-style: solid;
	border-right-color: #FFC4C4;
	border-bottom-color: #FFC4C4;
}
<!--   右侧表格样式   -->
.right {
	border-right-width: 1px;
	border-bottom-width: 1px;
	border-right-style: solid;
	border-bottom-style: solid;
	border-right-color: #FFC4C4;
```

```
        border-bottom-color: #FFC4C4;
}
<!--  文本框样式  -->
.txt {
        margin: 0px;
        padding: 0px;
        height: 18px;
        width: 100px;
        border: 1px solid #979797;
}
<!--  小文本样式  -->
.shorttxt {
        margin: 0px;
        padding: 0px;
        height: 18px;
        width:50px;
        border:1px solid #979797;
}
<!--  长文本样式  -->
.langtxt {
        margin: 0px;
        padding: 0px;
        height: 18px;
        width: 200px;
        border:1px solid #979797;
}
<!--  文本域样式  -->
.areatxt {
        margin: 0px;
        padding: 0px;
        border:1px solid #979797;
}
<!--  按钮样式  -->
.btn {
        font-size: 12px;
        color: #350407;
        background-color: #FFFFFF;
        margin: 0px;
        padding: 0px;
        height: 15px;
        width: 50px;
        border: 1px solid #000000;
}
<!--  整个表格样式  -->
.all{
        border:1px solid #979797;
}
```

本系统的公共文件介绍完毕，下面讲解本系统中重点模块的实现方法。



26.8　前台首页设计

视频讲解：光盘\TM\Video\26\前台首页设计.exe

前台首页一般没有多少实质性的技术，主要是加载一些功能模块，如登录模块、导航栏模块、公告栏模块等，使浏览者能够了解网站内容和特点。首页的重要之处是合理地对页面进行布局，既要尽可能地将重点模块显示出来，同时又不能因为页面凌乱无序，而让浏览者无所适从、产生反感。本系统前台首页 index.php 的运行结果如图 26.15 所示。

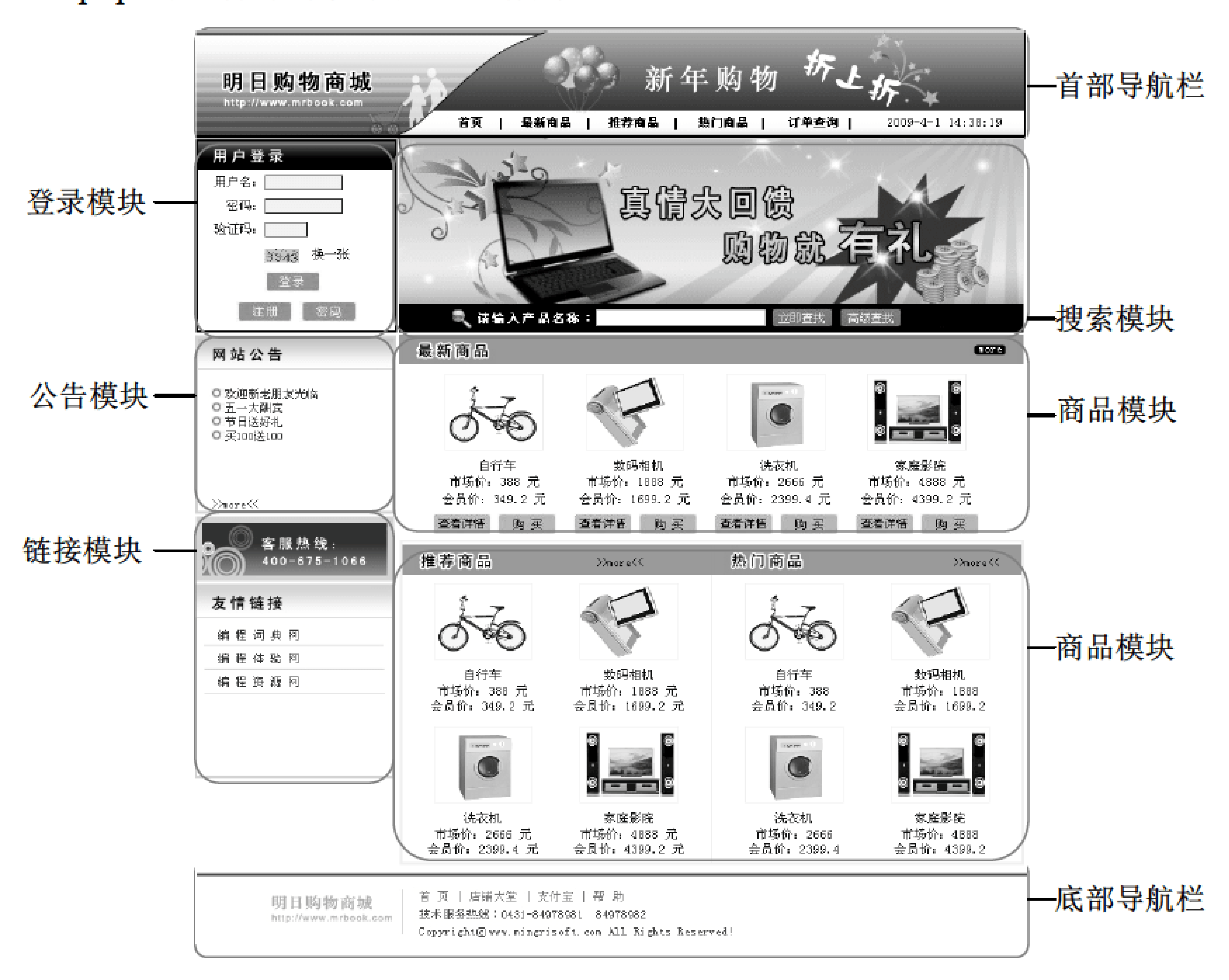


图 26.15　前台首页运行结果

本页面使用的是表格布局，除了商品模块和登录模块外，其他模块都是固定不变的。登录模块在会员成功登录后，将显示会员信息模块；而商品模块的位置，则根据不同的链接而显示不同的页面。index.php 页面的核心代码如下。

```
<?php
	session_start();/* 开启 Session	*/
?>
<table width="850" border="0" cellspacing="0" cellpadding="0">
  <tr>
      <!--	载入导航栏模块	-->
    <td colspan="2"><?php include_once 'top.php'; ?></td>
  </tr>
  <tr><td width="216" align="left" valign="top">
```

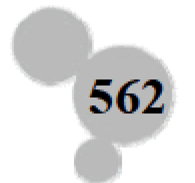
```
        <?php
            /*   载入登录模块、公告栏模块和友情链接模块   */
            include_once 'login.php';
            include_once 'public.php';
            include_once 'links.php';
        ?>
    </td>
    <td width="634" height="848" align="center" valign="top">
    <?php
            /*   载入查询模块   */
            include_once 'search.php';
        ?>
        <?php
        /*   根据不同的 page 值，载入不同的模块   */
    switch($_GET["page"]){
        case "":
            include_once 'nominate.php';                        //载入商品模块
            include_once 'newhot.php'; break;
        case "hyzx":include_once "member.php"; break;           //载入会员中心模块
        case 'allpub': include_once 'allpub.php'; break;        //显示公告栏模块
        case 'nom': include_once 'allnom.php'; break;           //显示推荐商品模块
        case 'new': include_once 'allnew.php'; break;           //显示最新商品模块
        case 'hot': include_once 'allhot.php'; break;           //显示热门商品模块
        case 'shopcar': include_once 'myshopcar.php' break;;    //显示购物车模块
        case 'settle': include_once 'settle.php'; break;        //收银台模块
        default:
            include_once 'nominate.php';                        //默认显示商品模块
            include_once 'newhot.php';
        }
        ?></td></tr>
</table>
<table width="850" border="0" cellspacing="0" cellpadding="0">
        <tr><td><?php include_once 'buttom.html'; ?></td></tr>    //底部导航栏模块
</table>
```

下面详细讲解其中几个重点模块的实现过程。只要将这几个模块所使用的技术和方法掌握好，其他模块的实现也就迎刃而解了。

说明： 本系统的功能较多，结构比较复杂，对于初学者来说学起来可能会比较困难。所以，本书将系统中的各个功能模块所涉及的文件（PHP、TPL、CSS、JS 等）尽可能都单独实现。读者在学习其中某个模块时，可以将相关的文件统一放到一个目录下单独测试。

26.9 登录模块设计

用户登录模块是会员进入的窗口。匿名用户虽然也可以访问本网站，但只能进行浏览、查询等简单操作，而会员则可以购买商品，并且能享受商品折扣。登录模块包括用户注册、用户登录和找回密



码 3 部分。

26.9.1　用户注册

视频讲解：光盘\TM\Video\26\用户注册.exe

用户注册通过一个 form 表单，将新用户填写的完整、合理、正确的信息保存到数据库中。如果用户填写的信息不正确或者不符合要求，则直接显示错误原因，以便用户修改。本系统的登录模块使用目前比较流行的 Ajax 技术来验证用户输入的数据，应用 GD2 函数库生成验证码。本页面的运行结果如图 26.16 所示。



图 26.16　注册模块页面

由于篇幅的限制，这里只给出有代表性的程序块，即用户名验证和验证码验证这两部分，而其他信息的验证方法在这两个程序块中都使用到了。

1．创建模板

首先创建模板文件。该模板主要包括注册表单和 div 标签。其中，div 标签是用来提示录入的回馈信息。模板文件 register.tpl 的核心代码如下。

```
<!--   载入 xmlhttp.js 文件和 check.js 文件  -->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/check.js"></script>
<!--   注册表单部分  -->
<form id="register" name="register" action="reg_chk.php" method="post" onSubmit="return chkinput(this)">
     <!--   用户名验证部分  -->
     <tr>
       <td width="81" height="25"><div align="right">用户名：</div></td>
       <td height="25" colspan="3"> 
           <!--   文本框，将会触发 onBlur 事件，调用 chkname()方法  -->
           <input id = " name " name =" name " type = " text " onBlur = " javascript:chkname( register ) " />
           <!--   隐藏域，默认值为“not”即不允许注册   -->
           <input id = " c_name " name = " c_anme " type = " hidden " value = " not " > <font color = " red " >
*</font></td>
```

```
            <!--    div 卷标用于显示对用户名的检测结果    -->
        <td height="25"><div id="name1"><font color="#999999">请输入用户名</font></div></td>
      </tr>
       <tr>
        <td height="25" colspan="2"> 
            <input type="submit" value="提交"/>
           
         <input type="reset" value="重写" /></td>
        <td height="25" colspan="3"><div style="color:#FF0000">带“*”号的为必填项</div></td>
      </tr>
   </form>
</table>
```

这里包含了两个 js 文件：createxmlhttp.js 和 check.js。其中，对 createxmlhttp.js 文件在 26.7 节中已作了详细的讲解，而 check.js 文件用于实现对用户名和验证码的验证过程。

2．实现用户名验证

用户名验证包括两个部分。第一部分是当用户输入名称后，焦点离开当前文本框时所触发的，触发的函数为 chkname(form)。代码如下。

```
/*   form 为传入的表单名称，本段代码为 register 表单    */
function chkname(form){
        /*   如果 name 文本域的信息为空，名为 name1 的 div 标签显示如下信息    */
        if(form.name.value==""){
                name1.innerHTML="<font color=#FF0000>请输入用户名！</font>";
        }else{
                /*   否则  获取文本域的值    */
                var user = form.name.value;
                /*   生成 url 链接，将 user 的值传到 chkname.php 页进行判断    */
                var url = "chkname.php?user="+user;
                /*   使用 XMLHttpRequest 技术运行页面    */
                xmlhttp.open("GET",url,true);
                xmlhttp.onreadystatechange = function(){
                if(xmlhttp.readyState == 4){
                                /*   根据不同的返回值，在 div 标签中输出不同信息    */
                                var msg = xmlhttp.responseText;
                                if(msg == '2'){
                                        name1.innerHTML="<font color=#FF0000>用户名被占用！</font>";
                                        return false;
                                }else if(msg == '1'){
                                        name1.innerHTML="<font color=green>恭喜您，可以注册!</font>";
                                /*   如果用户名正确，则将隐藏域的值改为“yes”    */
                                        form.c_name.value = "yes";
                                }else{
                                        name1.innerHTML="<font color=green>未知错误</font>";
                                }
                        }
                }
                xmlhttp.send(null);
```

```
        }
}
```

这个函数调用了 chkname.php 页，该页在会员登录时也会被调用，所以这里分两种情况：有密码和无密码。无密码为注册验证，当没有返回结果时，说明该用户名可用；有密码为登录验证，和无密码相反，只有查询记录存在时，才允许登录，并将用户名和用户 ID 存储到 Session 中。该页面代码如下：

```
<?php
session_start()                                              //开启 Session 支持
include_once "conn/conn.php";                                //加载数据库连接文件
$reback = '0';                                               //声明返回变数，初值为‘0’
$sql = "select * from tb_user where name='".$_GET['user']."'";   //生成 SQL 语句
/*    下面的 SQL 语句加上了对密码的比较，这是用来验证登录的，因为登录验证使用的也是本页面 */
$password = $_GET['password'];
if(!empty($password)){
        $sql .= " and password = '".md5($password)."'";          //如果密码不为空，说明是登录验证
}
/*    执行 SQL 语句  */
$rst = $conn->Execute($sql) or die('execute error');         //根据 name 字段查询数据库
if($rst->RecordCount() == 1){                                //判断 name 是否被使用
/*   登录所用    */
if($rst->fields['isfreeze'] != 0){
                $reback = '3';
        }else{
                $_SESSION['member'] = $rst->fields['name'];
                $_SESSION['id'] = $rst->fields['id'];
                $reback = '2';
        }
}else{
        $reback = '1';
}
echo $reback;
?>
```

上面的过程是第一次对用户名进行判断。当用户无视警告信息单击“提交”按钮时，就会触发<form>表单中的 onsubmit 事件，该事件触发 chkinput()函数，此函数通过隐藏域来判断该注册名称是否合法。程序代码如下：

```
function chkinput(form){
        if(form.name.value==""){                     //判断用户名称文本域是否为空
                form.name.focus();
                return false;
        }
        if(form.c_name.value == "not"){              //判断隐藏域 c_name 的值是否为 not
                form.name.select();                  //如果是 not，说明该名称未通过验证，返回 false
                return false;
        }
…
}
```

运行结果如图 26.17 所示。

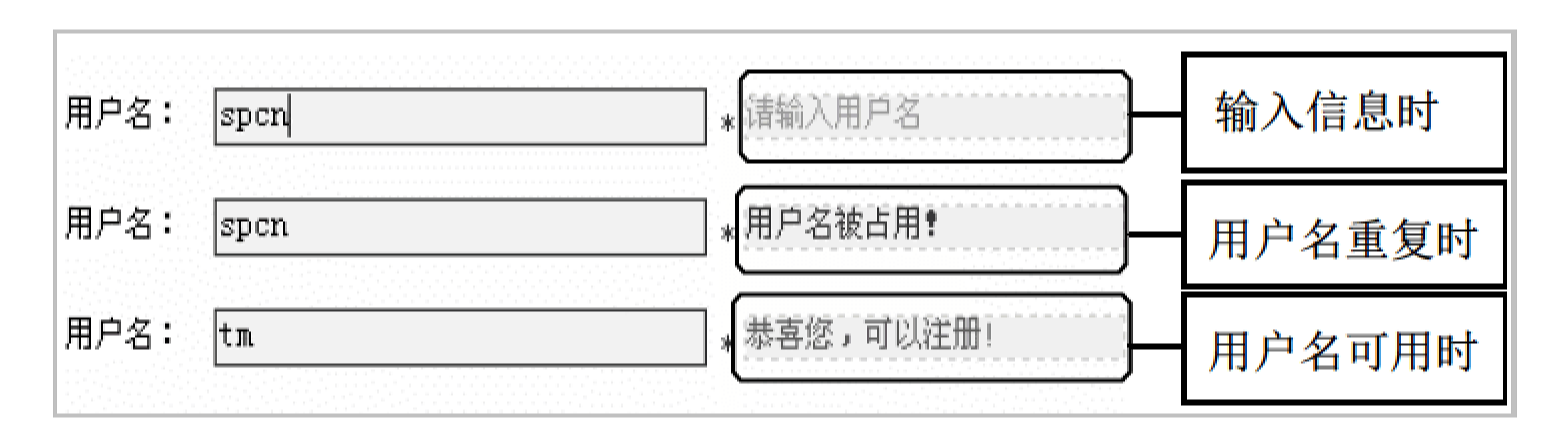


图 26.17　用户名验证过程

3．验证码验证

验证码验证，先来看 register.tpl 模板是如何显示验证码的，代码如下：

```
<!--   验证码验证部分    -->
        <tr>
          <td height="25"><div align="right">验证码：</div></td>
          <td width="65" height="25"> 
                <!--   验证码输入框，会触发 onBlur 事件    -->
              <input id="yzm" type="text" name="yzm" size="8" onBlur="javascript:chkyzm(register)" />
                <!--   隐藏域，保存验证码    -->
          <input name="yzm2" type="hidden" value="" /></td>
                <!--   验证码显示区    -->
          <td width="65" align="center" valign="middle"><script>yzm(register)</script></td>
                <!--   在看不清验证码的时候更换验证图片    -->
          <td width="51"><a href="javascript:code()">看不清</a></td>
                <!--   div 标签，用于显示验证信息    -->
          <td height="25"><div id="yzm1"><font color="#999999">输入验证码</font></div></td>
        </tr>
…
```

通过模板页代码可以看到，模板页通过调用 yzm()函数来显示验证码。yzm()函数的代码如下：

```
function yzm(form){
        var num1=Math.round(Math.random()*10000000);                                   //产生随机数
        var num=num1.toString().substr(0,4);                                           //获取前 4 位数
        document.write("<img name=codeimg src='yzm.php?num="+num+"'>");                //输出图片形式的验证码
        form.yzm2.value=num;                                                           //将验证码保存到隐藏域中
}
```

在 yzm()函数内，使用 Math.round()函数生成随机码，并截取前 4 位，然后将验证码提交给 yzm.php 页进行转换，并显示由 yzm.php 生成的 png 图片。最后将验证码存储到名字是 yzm2 的隐藏域中，用作验证码检查。yzm.php 页面使用的是 GD2 函数库。生成验证码的代码如下：

```
<?php
srand((double)microtime()*1000000);                                                    //生成随机数
$im=imagecreate(42,16);                                                                //创建画布
$black=imagecolorallocate($im,0,0,0);                                                  //定义背景
$white=imagecolorallocate($im,255,255,255);                                            //定义背景
$gray=imagecolorallocate($im,200,200,200);                                             //定义背景
```



```
	$password = md5($_POST['pwd1']);                          //使用 MD5 给密码加密
	$question = $_POST['question'];
	$answer = $_POST['answer'];
	$realname = $_POST['realname'];
	$card = $_POST['card'];
	$tel = $_POST['tel'];
	$phone = $_POST['phone'];
	$Email = $_POST['email'];
	$QQ = $_POST['qq'];
	$code = $_POST['code'];
	$address = $_POST['address'];
	$addtime = $conn->DBDate(time());                         //使用 DBDate()函数生成当前时间
	/***************************/
	/*  生成 SQL 语句  */
	$sql = "insert into tb_user ( name, password, question,answer, realname ,card, tel, phone, Email, QQ, code, address, addtime,isfreeze,shopping)" ;
	$sql .= " values ('$name', '$password', '$question', '$answer', '$realname', '$card', '$tel', '$phone', '$Email', '$QQ', '$code', '$address',$addtime,'0','')";
	/*  执行 SQL 语句  */
	$rst = $conn->execute($sql);
	/*  判断添加是否成功  */
	if($rst == false){
		echo '<script>alert(\'添加失败\');history.back;</script>';
	}else{
	/*  如果成功，则将用户 ID 和用户名称存到 Session 中  */
		$_SESSION['member'] = $name;
		$_SESSION['id'] = $conn->Insert_ID();            //Insert_ID()函数获取新增数据的 ID 号
		echo "<script>top.opener.location.reload();alert('注册成功');window.close();</script>";
	}
?>
```

5．加载模板

因为本系统使用了 Smarty，所以还需要创建一个 PHP 页来加载模板文件。register.php 页面的内容非常简单，代码如下：

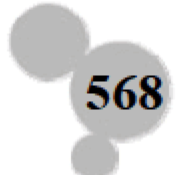
```
<?php
	include_once 'config.php';                     //载入 config.php 文件
	$smarty->assign('title','新用户注册');          //设置 title
	$smarty->display('register.tpl');              //显示 register.tpl
?>
```

6．创建链接

以上就是用户注册模块的核心代码。当测试没有问题后，最后创建一个“注册”超链接。当用户单击前台的注册按钮时，系统会调用 js 的 onclick 事件，弹出注册窗口。代码如下：

```
<a href="#" id="login" onclick="reg()">用户注册</a>
```

这里使用到的 js 文件为 js/login.js。调用的函数为 reg()，该函数代码如下：



```
function reg(){
    window.open("register.php", "_blank", "width=500,height=450",false);                    //弹出新窗口
}
```

26.9.2　用户登录

视频讲解：光盘\TM\Video\26\用户登录.exe

除了在注册时可以直接登录，其他情况下用户可以通过登录区进行登录操作。用户登录模块的运行结果如图 26.18 所示。



图 26.18　用户登录页面

1．信息验证

用户登录同样使用的是 Ajax 验证方法。首先创建页面模板文件 login.tpl，该文件主要包含一个登录表单。登录表单各元素的属性值及超链接说明如表 26.1 所示。

表 26.1　登录表单各元素属性说明

元 素 名 称	元 素 属 性	元 素 说 明
<form>	id = " login " name = " login " method = " post " action = " # " onsubmit = " return lg(this) "	登录表单可以触发 onsubmit 事件来调用 lg()函数，表单名称为 login
text	id="name" name="name" type="text"	用户名文本框
text	id="password" name="password" type="password"	密码文本框
submit	id="enter" name="enter" type="submit"	登录按钮
<a>看不清	onclick="javascript:code(login)" style=" cursor:hand"	更换验证码图片
hidden	name="check2" type="hidden" value=""	保存验证码的隐藏域
<a>用户注册	id="login" href="#" onclick="reg()"	注册页面超链接
<a>找回密码	id="login" href="#" onclick="found()"	找回密码超链接

通过表 26.1 可知，当单击 submit 按钮时，系统将调用 lg()函数。lg()函数包含在 js/login.js 脚本文件内，该函数代码如下：

```
function lg(form){
    /*    检验用户名      */
    if(form.name.value==""){
        alert('请输入用户名');
        form.name.focus();
        return false;
    }
    /*    检验密码   */
    if(form.password.value == "" || form.password.value.length < 6){
        alert('请输入正确密码');
        form.password.focus();
        return false;
    }
    /*    检验验证码      */
    if(form.check.value == ""){
```

```
            alert('请输入验证码');
            form.check.focus();
            return false;
        }
        /*    比较验证码    */
        if(form.check.value != form.check2.value){
            form.check.select();
            code(form);
            return false;
        }
        /*    根据输入的用户名和密码生成 url    */
        var user = form.name.value;
        var password = form.password.value;
        var url = "chkname.php?user="+user+"&password="+password;
        /*    使用 xmlhttp 对象来返回验证结果    */
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                /*    如果返回值不等于 2，说明用户名或密码错误    */
                if(msg == '1'){
                    alert('用户名或密码错误!!');
                    form.password.select();
                    form.check.value = '';
                    code(form);
                    return false;
                }if(msg == "3"){
                    alert("该用户被冻结，请联系管理员");
                    return false;
                } else{    /*    如果成功，重新刷新本页    */
                    alert('欢迎光临');
                    location.reload();
                }
            }
        }
        xmlhttp.send(null);
        return false;
}
```

用户名和密码是在 chkname.php 页面中被验证的。chkname.php 的相关内容在 26.9.1 节中已经介绍过，这里不再赘述。

2．用户信息

用户登录成功后，在原登录框位置将显示用户信息模块。用户可以通过“会员中心”超链接对自己的信息进行修改；也可以单击“查看购物车”超链接查看购物车商品；用户还可以通过单击“安全离开”超链接离开。关于会员信息模块，将在 26.10 节中进行讲解。用户信息模块的主要代码如下：

```
<!--    显示当前登录用户名    -->
欢迎您：{$member}
```



1．创建模板文件

虽然找回密码需要 4 个步骤，但实际上每个步骤使用的都是相同的模板文件和 js 文件，只是被调用的表单和 js 函数略有差别。这里根据不同的文件分别进行介绍。

首先来看模板文件。该模板文件一共包含了 3 个表单，分别代表了 3 个步骤。下面是模板文件的核心代码。

```
<!-- 载入两个 js 脚本文件 -->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/found.js"></script>
<!-- 第一个 div 标签 -->
<div id="first">
<table width="200" border="0" cellspacing="0" cellpadding="0">
<form id="foundname" name="found" method="post" action="#">
  <tr><td> 找回密码</td></tr>
  <tr><td>会员名称：</td>
     <!-- text 文本域，用于输入要找回密码的会员名称 -->
      <td><input id="user" name="user" type="text" class="txt"></td>
  </tr>
  <tr><td>
<!-- 单击“下一步”按钮，能触发 onclick 事件来调用 chkname()函数 -->\
<input id = " next1 " name = " next1 " type = " button " class = " btn " value = " 下一步 " onClick = " return chkname
( foundname ) "/></td></tr>
</form>
</table>
</div>
<!-- 第二个 div 标签，样式为隐藏 -->
<div id="second" style="display:none;">
<table>
<form id="foundanswer" name="found" method="post" action="#">
  <tr><td > 找回密码</td></tr>
  <tr><td>密保问题：</td>
<!-- 用于显示密保问题的 div 标签 -->
    <td <div id="question"></div></td></tr>
  <tr><td>密保答案：</td>
<!-- 文本域，用于填写密保答案 -->
    <td ><input id="answer" name="answer" type="text" class="txt" /></td></tr>
  <tr>
<!-- 单击“下一步”按钮，用来触发 onclick 事件，并调用 chkanswer()函数 -->
    <td><input id = " next2 " name = " next2 " type=" button " class=" btn " value =" 下一步 " onClick = " return
chkanswer( foundanswer ) "></td>
    </tr>
</form>
</table>
</div>
<!-- 第三个 div 标签，样式也为隐藏，作用是修改密码 -->
<div id='third' style="display:none;">
<table>
<form id="modifypwd" name="found" method="post" action="#">
```

```
<tr><td> 输入密码</td></tr>
<tr><td>输入密码：</td>
  <td><input id="pwd1" name="pwd1" type="password" class="txt"></td></tr>
<tr><td>确认密码：</td>
  <td><input id="pwd2" name="pwd2" type="password" class="txt" /></td>
</tr>
<tr>
<!--  单击“完成”按钮，调用 chkpwd()函数  -->
  <td><input id = " mod " name = " mod " type = " button " class = " btn " value = "  完成  " onClick = " return chkpwd
(modifypwd) "></td>
  </tr>
</form>
</table>
</div>
```

可以发现，3 个表单中，只有一个表单默认情况下是显示的，其他则为隐藏。只有通过调用不同的 js 函数，才可以对其他表单进行操作。

2．创建 js 脚本文件

found.js 脚本文件包含 3 个函数：chkname()、chkanswer()和 chkpwd()。其中 chkname()函数的作用是检查用户输入的会员名称，如果存在，则使用 xmlhttp 对象去调用生成的 url 进行判断。如果该用户存在，则隐藏当前表单，并显示下一个表单，最后输出密保问题。chkname()函数的代码如下：

```
function chkname(form){
	/*	获取文本框信息	*/
	var user = form.user.value;
	/*	如果为空，则输出提示	*/
	if(user == ''){
		alert('请输入用户名');
		form.user.focus();
		return false;
	}else{
	/*	否则，生成 url，并调用 xmlhttp 对象	*/
		var url = "foundpwd.php?user="+user;
		xmlhttp.open("GET",url,true);
		xmlhttp.onreadystatechange = function(){
		if(xmlhttp.readyState == 4){
				var msg = xmlhttp.responseText;
				/*	如果没有结果，则提示	*/
				if(msg == '0'){
					alert('没有该用户，请重新查找!');
					form.user.select();
					return false;
				}else{
				/*	否则，隐藏当前表单，显示下一个表单，并输出密保问题 */
					document.getElementById('first').style.display = 'none';
					document.getElementById('second').style.display = '';
					document.getElementById('question').innerHTML = msg;
				}
```

```
                }
            }
            xmlhttp.send(null);
        }
}
```

其他两个函数也使用了 XMLHttpRequest 对象，实现方法相差无几，不同之处在于对返回值的处理。chkanswer()函数隐藏当前表单，显示下一个表单。chkanswer()函数的代码如下：

```
function chkanswer(form) {
        /*      获取已隐藏表单数据 */
        var user = document.getElementById('user').value;
        /*      获取密保答案    */
        var answer = form.answer.value;
        /*      表单验证    */
        if(answer == ''){
                ...
        }else{
                /*      生成 url     */
                var url = "foundpwd.php?user="+user+"&answer="+answer;
                xmlhttp.open("GET",url,true);
                xmlhttp.onreadystatechange = function(){
                        if(xmlhttp.readyState == 4){
                                var msg = xmlhttp.responseText;
                                if(msg == '0'){
                                        alert('问题回答错误');
                                        form.answer.select();
                                        return false;
                                }else{
                                        document.getElementById('second').style.display = 'none';
                                        document.getElementById('third').style.display = '';
                                }
                        }
                }
                xmlhttp.send(null);
        }
}
```

而 chkpwd()函数则弹出提示信息，如果成功，则关闭当前页。chkpwd()函数的代码如下：

```
function chkpwd(form){'
        /*      获取名称和密码        */
        var user = document.getElementById('user').value;
        var pwd1 = form.pwd1.value;
        var pwd2 = form.pwd2.value;
        /*      省略部分为表单验证    */
        ...
        var url = "foundpwd.php?user="+user+"&password="+pwd1;
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
```

```
          if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                if(msg == '1'){
                     alert('密码修改成功，请重新登录');
                     window.close();
                }
          }
     }
     xmlhttp.send(null);
}
```

3．创建数据处理文件

foundpwd.php 文件的功能是根据用户输入信息来检测数据表中的数据，并根据不同的输入项返回不同的结果，代码如下。

```
<?php
     include_once 'conn/conn.php';                                    //载入数据库连接文件
     $name= $_GET['user'];                                            //获取表单值
     $answer = $_GET['answer'];
     $password = $_GET['password'];
     $reback = '0';
     /*    如果变量$answer 和$password 为空，说明用户只输入了会员名称    */
     if(empty($answer) && empty($password)){
          /*    生成查询语句    */
          $namesql = "select * from tb_user where name = '".$name."'";
          /*    获取查询结果    */
          $namerst = $conn->execute($namesql);
          /*    如果查询记录为 1，说明输入正确    */
          if($namerst->recordCount() == 1){
               /*    获取密保问题，并赋给变量$reack    */
               $question = $namerst->fields['question'];
               $reback = $question;
          }
     /*    如果变量$answer 不为空，说明用户输入了密保答案    */
     }else if(!empty($answer)){
          /*    根据密保答案和会员名称，生成查询语句   */
          $answersql = "select * from tb_user where name = '".$name."' and answer = '".$answer."'";
          /*    返回查询结果，并判断是否有返回记录，如果有，说明输入正确    */
          $answerrst = $conn->execute($answersql);
          if($answerrst->recordCount() == 1){
               $reback = '1';
          }
     /*    如果变量$password 不为空，说明用户输入了新密码   */
     }else if(!empty($password)){
          /*    根据用户名称和新密码更新记录    */
          $sql = "select * from tb_user where name = '".$name."'";
          $arr = array();
          $rst = $conn->execute($sql);
          if($rst->RecordCount() == 1){
```

```
                    $arr['password'] = md5($password);
                    $updateSQL = $conn->GetUpdateSQL($rst,$arr,true);
                    $conn->execute($updateSQL);
                    $reback = '1';
               }
          }
          /*    输出返回值      */
          echo $reback;
?>
```

4．加载模板页

因为所有登录模块的模板都是不需要、或者只需要传递一两个变量，所以 PHP 加载页的内容比较简单。找回密码页面的 PHP 加载文件是 found.php，代码可参考配套光盘中的内容。

26.10　会员信息模块设计

视频讲解：光盘\TM\Video\26\会员信息模块设计.exe

用户登录后，即可看到会员信息模块。在会员信息模块中，可以执行查看、修改个人信息及密码，查看购物车和安全退出等操作。本节只对会员信息模块中的“会员中心”和“安全退出”超链接进行讲解，关于“查看购物车”超链接将在商品模块中进行介绍。

26.10.1　会员中心

单击“会员中心”超链接，会回传给当前页一个 page 值，当前页根据这个 page 值来载入 member.php 文件。member.php 页面的运行结果如图 26.21 所示。



图 26.21　“会员中心”页面的运行结果

1．创建 PHP 页面

与登录模块设计不同，本节首先来创建 PHP 页面，因为该模块中的模板需要使用数据库中的数据



及一些动态信息，这些都需要在 PHP 页中先行获取及处理，然后再传给模板页。“会员中心”页面的代码如下。

```
<?php
    session_start();                                          //开启 Session 支持
    include_once 'config.php';                                //载入 Smarty 配置文件
    include_once 'conn/conn.php';                             //载入数据库连接文件
    /*    查找用户资料    */
    $sql = 'select * from tb_user where id = '.$_SESSION['id'];
    $rst = $conn->execute($sql);
    $arr = $rst->GetArray();                                  //将结果集输出为数组
    if($_GET['action'] == 'modify'){                          //如果动作为修改密码
            $smarty->display('modifypwd.tpl');                //则载入修改密码模板页
    }else{                                                    //否则载入查看信息页
            $smarty->assign('arr',$arr[0]);                   //传递数组
            $smarty->display('membershow.tpl');               //显示模板
    }
?>
```

2．创建模板页

该模块包括查看信息模板及修改密码模板。首先介绍查看信息模板，代码如下：

```
<!--  显示超链接  -->
<p>{$smarty.session.member}<a href = ' ?page=hyzx ' >查看信息</a><a href = ' ?page = hyzx&action=modify' id="mem">修改密码</a></p>
<table>
<form id = " member " name = " member " method = " post " action = " modify_info_chk.php " onsubmit = " return mem( member ) " >
<!--  显示当前会员信息，此处为不可更改信息 -->
  <tr><td colspan="2">{$arr.name}信息（不可更改信息）</td></tr>
<!--  因为格式相同，所以只显示一条，其他省略  -->
  <tr><td>会员编号：</td><td > {$arr.id}</td></tr>
    …
  <tr><td>{$arr.name}信息（可更改信息）</td></tr>
<!--  可以修改的信息，放到表单元素中  -->
  <tr><td>真实姓名：</td>
    <td><input id="realname" name="realname" type="text" value="{$arr.realname}" /></td></tr>
<!--  省略了部分代码……  -->
  <tr><td>
     <input name="enter" type="submit" id="enter" value="修改" />
     <input name="reset" type="reset" id="reset" value="重置" />
  </td></tr>
</form>
</table>
```

修改密码模板比信息模板要简单一些，因为不需要额外的传值。模板代码如下：

```
<!--  载入脚本文件  -->
<script language="javascript" src="js/member.js"></script>
```

```
<p>{$smarty.session.member}<a  href='?page=hyzx'>查看信息</a><a  href='?page=hyzx&action=modify'  >修改密码
</a></p>
<table>
<!--   修改密码表单   -->
<form id = " member " name = " member " method = " post " action = " modify_pwd_chk.php " onsubmit = " return
pwd(member) ">
   <tr><td >修改密码</td></tr>
   <tr><td >原密码：</td><td><input id="old" name="old" type="password" /></td></tr>
   <tr><td>新密码：</td><td><input id="new1" name="new1" type="password" /></td></tr>
   <tr><td>确认密码：</td><td><input id="new2" name="new2" type="password" /></td></tr>
   <tr><td><input id="enter" name="enter" type="submit" value="修改" /></td></tr>
</form>
</table>
```

3．创建脚本文件

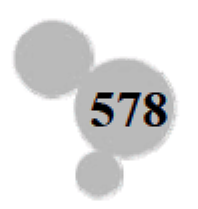
该模块的脚本文件和用户注册模块类似，都是对信息的合法性进行验证，如信息是否为空、是否符合规范等。这里不给出代码，读者可以参考 26.9.1 节的内容。

4．创建处理页

当信息验证通过后，系统将跳到处理页，进行信息处理。本模块的处理页分为信息修改和密码修改两个页面。首先介绍信息修改页，代码如下：

```
<?php
	session_start();                                                   //开启 Session 支持
	include_once 'conn/conn.php';                                      //载入数据库连接文件
	$sql = 'select * from tb_user where id = '.$_SESSION['id'];        //根据 ID 生成 SQL 查询语句
	$rst = $conn->execute($sql);                                       //执行 SQL 语句，返回查询结果集
	$mod = array();                                                    //声明数组
	/*     将修改信息添加到数组中   */
	$mod['realname'] = $_POST['realname'];
	$mod['card'] = $_POST['card'];
	$mod['tel'] = $_POST['tel'];
	$mod['phone'] = $_POST['phone'];
	$mod['Email'] = $_POST['email'];
	$mod['QQ'] = $_POST['qq'];
	$mod['code'] = $_POST['code'];
	$mod['address'] = $_POST['address'];
	/****************************************/
	$updateSQL = $conn->GetUpdateSQL($rst,$mod);                      //生成更新语句
	if($conn->execute($updateSQL))
		echo "<script>alert('修改成功');location=('index.php');</script>";   //修改成功，跳到首页
	else
		echo "<script>alert('修改失败');history.go(-1);</script>";          //否则返回上一页
?>
```

密码修改页的操作流程与信息修改页类似，只是更新的数组少，只有一个字段。修改密码页的代码如下：



```
<?php
	session_start();                                                            //开启 Session 支持
	include_once 'conn/conn.php';                                              //载入数据库连接文件
	$oldpwd = md5($_POST['old']);                                              //为旧密码进行 MD5 加密
	/*  根据 ID 和 MD5 加密后的密码查询用户              */
	$sql = 'select * from tb_user where id = '.$_SESSION['id'].' and password = \''.$oldpwd.'\'';
	$rst = $conn->execute($sql);                                               //返回查询结果
	$mod = array();                                                            //创建更新数组
	$mod['password'] = md5($_POST['new1']);                                    //为新密码进行 MD5 加密
	$updateSQL = $conn->GetUpdateSQL($rst,$mod);                               //生成更新语句
	if($conn->execute($updateSQL))
		echo "<script>alert('修改成功');location=('index.php');</script>";     //修改成功，返回首页
	else
		echo "<script>alert('修改失败');history.go(-1);</script>";             //否则返回上一页
?>
```

26.10.2　安全退出

当用户需要离开网站时，可以单击“安全退出”超链接来调用 logout()函数，当确认用户退出后，跳转到 logout 页面，销毁 Session，并回到首页。安全退出所涉及的页面及代码如下：

```
function logout(){
	if(confirm("确定要退出登录吗？ ")){                   //输出选择框，用户可以单击“确认”或“取消”按钮
		window.open('logout.php','_parent','',false);    //如果用户确认退出，则打开 logout.php 页
	}else
	 return false;
}
<?php
	session_start();                                                              //开启 Session 支持
	session_destroy();                                                            //销毁 Session
	echo '<script>alert(\'用户已安全退出!\');location=(\'index.php\');</script>';   //回到首页
?>
```

26.11　商品展示模块

视频讲解：光盘\TM\Video\26\商品展示模块.exe

本系统根据商品的类别，为用户提供了多种不同的商品展示方式，包括推荐商品、最新商品和热门商品等，能够使消费者有目的地选购一些商品。商品信息对商品的一些属性进行了描述，还提供了商品的实物图片，为用户提供信息，让用户对商品有更多的了解。商品展示模块对电子商务平台来说尤为重要。本系统中商品展示模块的运行结果如图 26.22 所示。

因为推荐商品、最新商品和热门商品的实现方法和过程基本相同，所以本节只讲解推荐商品模块，其他的程序可参见本书配套光盘。

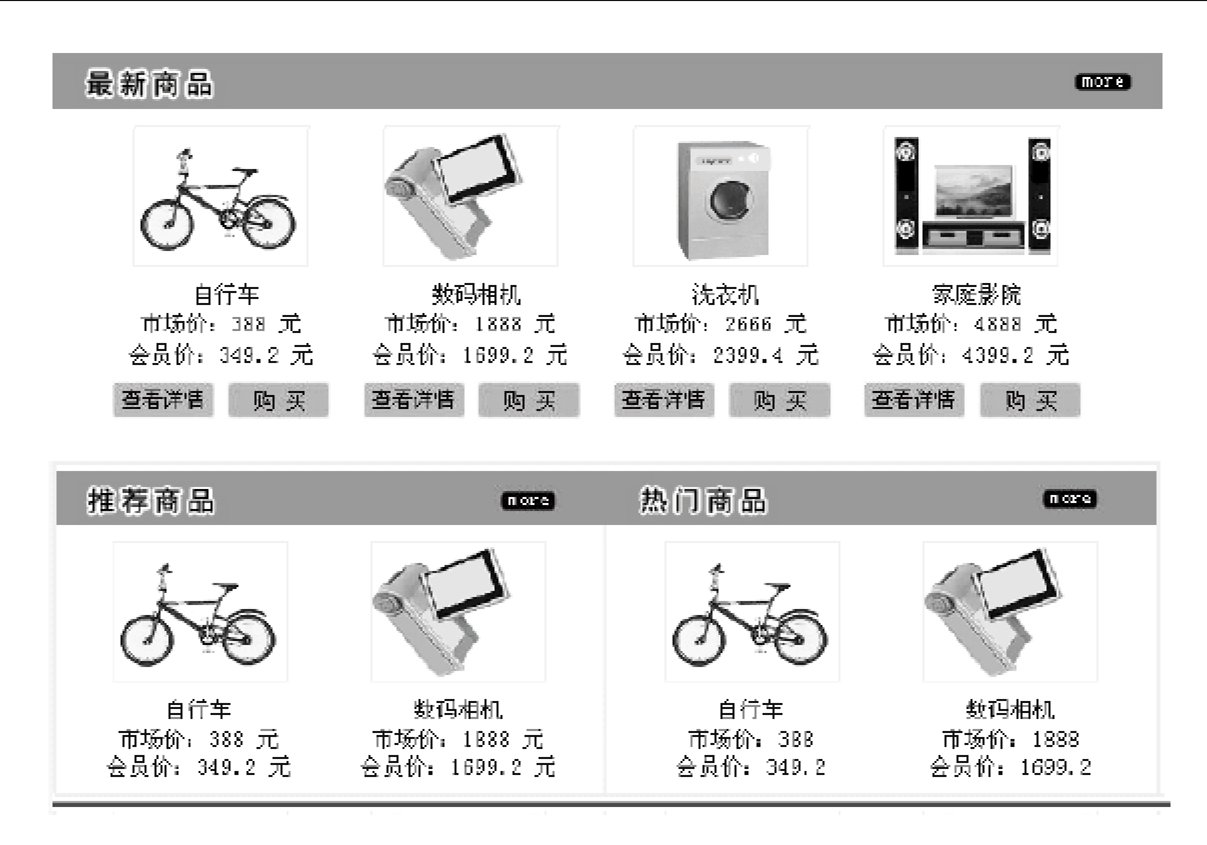


图 26.22　商品展示模块页面

26.11.1　创建 PHP 页

既然是商品展示模块，首先就是要取得商品记录。在数据库设计一节中已经介绍，商品字段 isnom 表示商品是否被推荐，如果该字段为 1，即为推荐，否则为不推荐。所以在 PHP 页面中，首先要获取数据表中 isnom 字段属性为 1 的记录，因为受页面限制，所以只返回 4 条记录。首页中的推荐商品页面的代码如下：

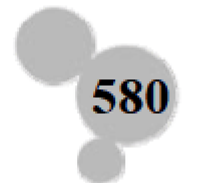
```
<?php
    include_once 'conn/conn.php';                                   //载入数据库连接文件
    include_once 'config.php';                                      //载入模板配置文件
    /*    根据 isnom 字段，生成 SQL 语句      */
    $sql = "select id,name,pics,m_price,v_price from tb_commo where isnom = 1";
    $num = 4;                                                       //设置提取记录条数
    $rst = $conn->SelectLimit($sql,$num);                           //使用 SelectLimit 方法获取记录集
    $nomarr = $rst->GetArray();                                     //将记录集保存为数组

    $smarty->assign('nomarr',$nomarr);                              //将数组传递到模板页
    $smarty->display('nominate.tpl');                               //显示模板
?>
```

26.11.2　创建模板页

记录集数组被传递到模板页后，模板页使用 foreach 标签输出商品信息，并添加相应的操作按钮和超链接。模板页中一共有显示更多商品、查看商品和放入购物车 3 个事件。

- ☑ 当单击“更多商品”超链接时，将重新加载本页面，并传递一个 page 变量。switch 语句会根据 page 值来显示。
- ☑ 单击“查看商品”按钮时，将触发 onclick 事件，并调用 openshowcommo()函数，同时，商品 ID 会作为函数的唯一参数被传递。
- ☑ 当单击“放入购物车”按钮时，同样会触发 onclick 事件，并调用 buycommo()函数，唯一的



参数也是商品的 ID。

商品模板页面的代码如下：

```
<!--   载入 js 脚本文件   -->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/showcommo.js"></script>
<table border="0" cellpadding="0" cellspacing="0">
	<!--   “更多商品”超链接   -->
	<tr><td><a href="?page=new">&gt;&gt;more&lt;&lt;</a></td><td rowspan="3"> </td></tr>
	<tr><td > </td>
	<!--   使用 Smarty 的 foreach 标签，循环输出数组   -->
	    {foreach key=key item=item from=$nomarr}
	   <td>
	       <table border="0" cellpadding="0" cellspacing="0" >
	              <tr><td>
	              <!--   商品图片显示   -->
	                     <img src="{$item.pics}" width="100" height="80" alt="{$item.name}"></td></tr>
	              <tr><td> {$item.name}</td></tr>
	                 <tr><td>市场价：{$item.m_price} 元</td></tr>
	                 <tr><td>会员价：{$item.v_price} 元</td></tr>
	                 <tr><td>
	              <!--   “查看商品”按钮   -->
	                     <input id = " showinfo " name = " showinfo " type = " button " value = "" onclick = " openshowcommo
({$item.id})"/>
	              <!--   “放入购物车”按钮   -->
	                     <input id = " buy " name = " buy " type = " button " value = "" onclick = " return buycommo
( {$item.id} ) " /></td></tr></table></td>
	<!--   循环结束   -->
	  {/foreach}
	     <td> </td></tr>
	<tr><td colspan="6"> </td></tr>
</table>
```

26.11.3　js 脚本页面

在模板页中的 3 个链接，其中两个都用到了 js 函数。通过 js 函数，当单击“查看商品”按钮时，系统会弹出一个新的页面，并显示商品的详细信息；当单击“购买”按钮时，该商品将会被放到当前用户的购物车中，如果没有登录用户或商品已添加，则弹出提示信息。js 脚本文件的代码如下：

```
/*   查看商品信息函数，将打开一个新页面   */
function openshowcommo(key){
	open('showcommo.php?id='+key,'_blank','width=560 height=300',false);
}
/*   将购买商品添加到购物车中，将在 26.12 节中进行讲解   */
function buycommo(key){
	…
}
```

显示更多商品列表和商品详细信息页的实现过程和 26.11.1 节、26.11.2 节所述类似，只是显示更多字段信息而已，这里不再赘述。26.12 节将重点讲解购物车的实现。

26.12　购物车模块设计

视频讲解：光盘\TM\Video\26\购物车模块设计.exe

购物车在电子商务平台中是非常关键的一个功能模块。购物车保留用户选择的商品信息，用户可以在购物车内设置选购商品的数量，显示选购商品的总金额，清除选择的全部商品，以及重新选择商品。购物车页面运行结果如图 26.23 所示。

我的购物车

	商品名称	购买数量	市场价格	会员价格	折扣率	合计
□	自行车	3	388	349.2	9	1047.6
□	数码相机	5	1888	1699.2	9	8496
□	洗衣机	1	2666	2399.4	9	2399.4
□	家庭影院	1	4888	4399.2	9	4399.2

全选 反选 删除选择 继续购物 去收银台 共计：16342.2 元

图 26.23　购物车页面运行结果

购物车模块主要实现以下功能：添加商品、删除商品、更改商品数量。

26.12.1　添加商品

在商品展示模块中，单击指定商品中的“购买”按钮，即可将物品放到购物车中，并进入到“购物车”页面。从 26.11 节中已经了解到“购买”动作调用的是 buycommo()函数，购买商品的 ID 是该函数的唯一参数。下面就来看 buycommo()函数的功能结构。此函数代码如下：

```
/*添加商品，同时检查用户是否登录、商品是否重复等	*/
function buycommo(key){
	/*	根据商品 ID，生成 url	*/
	var url = "chklogin.php?key="+key;
	/*	使用 xmlhttp 对象调用 chklogin.php 页	*/
	xmlhttp.open("GET",url,true);
	xmlhttp.onreadystatechange = function(){
		if(xmlhttp.readyState == 4){
				var msg = xmlhttp.responseText;
				/*	用户没有登录	*/
				if(msg == '2'){
					alert('请您先登录');
					return false;
				}else if(msg == '3'){
				/*	商品已添加	*/
					alert('该商品已添加');
					return false;
				}else{
				/*	显示购物车	*/
```

```
                    location='index.php?page=shopcar';
                }
            }
    }
    xmlhttp.send(null);
```

可以看到，buycommo()函数通过 xmlhttp 对象调用 chklogin.php 页，并根据回传值作出相应的处理，chklogin.php 页才是将商品添加到购物车中的操作页。chklogin.php 页的代码如下：

```
<?php
    session_start();                                              //开启 Session 支持
    /**
     *  1 表示添加成功
     *  2 表示用户没有登录
     *  3 表示商品已添加过
     *  4 表示添加时出现错误
     *  5 表示没有商品添加
     */
    include_once 'conn/conn.php';                                 //载入数据库连接文件
    $reback = '0';                                                //声明返回值
    if(empty($_SESSION['member'])){                               //如果 Session 值为空，说明没有登录
        $reback = '2';                                            //参见头注释
    }else{
        $key = $_GET['key'];                                      //获取商品 ID
        if($key == ''){
            $reback = '5';
        }else{
            $id = (int)$_SESSION['id'];                           //获取登录会员 ID 号
            $boo = false;                                         //设置一个布尔变量，初始为 false
            $addshop = array();                                   //数据库更新数组
            $sql = "select id,shopping from tb_user where id = ".$id;  //生成当前用户的查询语句
            $rst = $conn->execute($sql);                          //返回当前用户记录集
            $shopcont = $rst->fields['shopping'];                 //找到 shopping 字段
            if(!empty($shopcont)){                                /如果字段不为空，说明已有商品
                $arr = explode('@',$shopcont);                    //将字段用“@”拆分并另存数组
                foreach($arr as $value){                          //循环输出数组内容
                    if($key == $value[0]){                        //判断需要添加的商品是否存在
                        $reback = '3';                            //参见头注释
                        $boo = true;                              //同时布尔变量变为 true
                        break;                                    //停止循环
                    }
                }
                if(!$boo){                                        //如果布尔值没有改变，再添加商品
                    /*  将商品数量默认设为 1，连同商品 ID 保存到字串中  */
                    $shopcont .= '@'.$key.',1';
                    /*  将商品字串保存到更新数组中  */
                    $addshop['shopping'] = $shopcont;
                    /*  生成更新语句  */
                    $updateSQL = $conn->GetUpdateSQL($rst,$addshop);
                    /*  更新数据表  */
```

```
        /*    如果有结果集，则将结果集另存为数组    */
        $tmparr = $rst->GetAssoc();
        $commarr = array();
        /*    输出 shopping 字段值            */
        foreach($tmparr as   $value){
            /*    使用“@”进行分隔，得到数组*/
            $tmpnum = explode('@',$value);
            /*    得到购买商品种类    */
            $shopnum = count($tmpnum);
            /*    商品总价值      */
            $sum = 0;
            /*    数组循环  */
            foreach($tmpnum as $key => $vl){
                /*    使用逗号“,”拆分单一的商品信息          */
                $s_commo = explode(',',$vl);
                /*    根据保存的商品信息查找商品    */
                $sql2 = "select id,name,m_price,fold,v_price from tb_commo";
                $commsql = $sql2." where id = ".$s_commo[0];
                /*    返回结果集      */
                $commrst = $conn->execute($commsql);
                /*    将结果集另存数组    */
                $arr = $commrst->GetArray();
                /*    将需要显示的信息处理后保存到数组中        */
                $arr[0]['num'] = $s_commo[1];
                $arr[0]['total'] = $s_commo[1]*$arr[0]['v_price'];
                $sum += $arr[0]['total'];
                $commarr[$key] = $arr[0];
            }
        }
        /*    将得到的两个数组和总消费金额传递给模板      */
        $smarty->assign('shoparr',$shopnum);
        $smarty->assign('commarr',$commarr);
        $smarty->assign('sum',$sum);
        $smarty->assign('title','我的购物车');
        /*    显示购物车模板      */
        $smarty->display('myshopcar.tpl');
?>
```

商品的模板页不仅要负责将传入的商品信息数组显示出来，而且还要提供可以对商品进行修改、删除等操作的接口。模板页的代码如下：

```
<!--   载入使用到的 js 脚本文件   -->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/shopcar.js"></script>
<table>
<!--   购物车表单   -->
<form id="myshopcar" name="myshopcar" method="post" action="#">
  <tr><td>我的购物车</td></tr>
  <tr><td > </td><td>商品名称</td><td>购买数量</td><td>市场价格</td>
     <td>会员价格</td><td>折扣率</td><td>合计</td></tr>
```

```
<!--  使用 foreach 表单循环输出商品种类  -->
{foreach key=key item=item from=$commarr}
<!--  删除复选框，名称用数组表示，每个复选框的值为商品的 ID 号  -->
  <tr><td><input id="chk" name="chk[]" type="checkbox" value="{$item.id}"></td>
<!--  显示该商品的名称  -->
          <td><div id = "c_name{$key}">  {$item.name}</div></td>
    <td>
<!--  商品数量文本框 每个复选框的名称为“cnum”+key 值  -->
          <input id = " cnum{$key} " name = " cnum{$key} " type = " text " value = " {$item.num} " onkeyup = "
cvp( {$key},{$item.v_price},{$shoparr})"></td>
<!--  商品的其他信息  -->
    <td><div id="m_price{$key}"> {$item.m_price}</div></td>
    <td><div id="v_price{$key}"> {$item.v_price}</div></td>
    <td><div id="fold{$key}"> {$item.fold}</div></td>
    <td><div id="total{$key}"> {$item.total}</div></td>
  </tr>
{/foreach}
  <tr>
    <td colspan="3">
<!--  全选、反选和删除操作  -->
       <a href="#" onclick="return alldel(myshopcar)">全选</a>
       <a href="#" onclick="return overdel(myshopcar);">反选</a>
       <input type="button" value="删除选择" onClick = 'return del(myshopcar);'></td>
    <td>
<!--  继续购物  -->
         <input type="button" value="继续购物" onclick="return conshop(myshopcar)" /></td>
    <td>
<!--  将当前用户名保存到隐藏域中  -->
         <input id="uid" name="uid" type="hidden" value="{$smarty.session.member}" >
<!--  结算，去收银台  -->
         <input type="button" class="btn" value="去收银台" onclick="return formset(form)" /></td>
    <td colspan="2"><div id='sum'>共计：{$sum} 元</div></td>
    </tr>
</form>
</table>
```

26.12.3 更改商品数量

对于新添加的商品，默认的购买数量为 1，在购物车页面可以对商品的数量进行修改。当商品数量发生变化时，商品的“合计”金额和商品总金额会自动发生改变。该功能是通过触发 text 文本域的 onkeyup 事件调用 cvp()函数实现的。cvp()函数有 3 个参数：商品 ID、商品单价和商品类别。

首先通过商品 ID 得到要修改商品的相关表单和标签属性，然后通过商品单价和输入的商品数量计算该商品的合计金额，接着使用 for 循环得到其他商品的合计金额，最后将所有的合计金额累加，并输出到购物车页面。cvp()函数的代码如下：

```
/*  自动刷新总金额
 * key：商品 ID
```



```
 * vpr：商品单价
 * shoparr：商品种类数
*/
function cvp(key,vpr,shoparr){
	var n_pre = 'total';                                    //div 标签的 ID 前缀
	var num = 'cnum'+key.toString();                        //根据 key 值生成文本域的 ID 值
	var total = n_pre+key.toString();                       //根据 key 值生成 div 标签的 ID 值
	var t_number = document.getElementById(num).value;      //获取输入的商品数量
	var ttl = t_number * vpr;                               //根据商品数量和单价，计算商品金额
	document.getElementById(total).innerHTML = ttl;         //更新商品总金额
	var sm = 0;                                             //所有商品的总金额，初始为 0
	for(var i = 0; i < shoparr; i++){                       //根据商品种类数，循环得到其他商品金额
		var aaa = document.getElementById(n_pre+i.toString()).innerText;
		sm += parseInt(aaa);                                //将所有商品金额累加
	}
	document.getElementById('sum').innerHTML = '共计：'+sm+' 元';
}
```

这里所更改的商品数量并没有被保存到数据库中，如果希望保存，则单击“继续购物”按钮，即可将商品数量更新到数据库中。该功能将在 26.12.5 节中实现。

26.12.4　删除商品

当对添加的商品不满意时，可以对商品进行删除操作。操作流程为：首先选中要删除的商品前面的复选框，如果全部删除，则可以单击“全选”按钮，或“反选”按钮；然后单击“删除选择”按钮；在弹出的警告框中单击“确定”按钮，商品将被全部删除。删除商品的操作流程如图 26.24 所示。

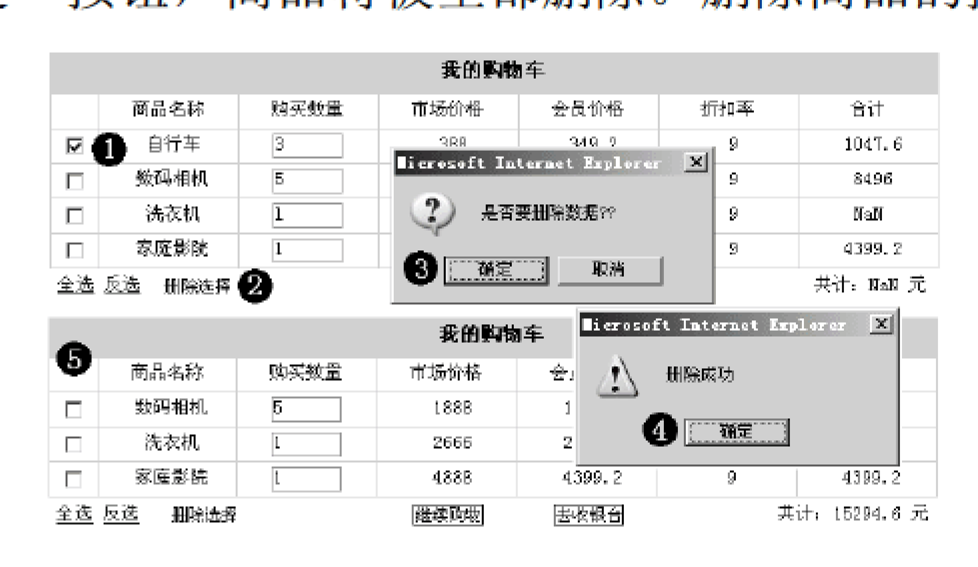


图 26.24　删除商品流程

所有的删除操作都是通过 js 脚本文件 shopcar.js 来实现的，相关的函数包括 alldel()函数、overdel()函数和 del()函数。下面分别介绍这 3 个函数的实现过程。

alldel()函数和 overdel()函数实现的原理比较简单，通过触发 onclick 事件来改变复选框的选中状态。实例代码如下：

```
/*	全部选择	*/
function alldel(form){
	var leng = form.chk.length;                  //获取复选框数量
	if(leng==undefined){                         //如果等于 undefined，说明只有一个复选框
	    if(!form.chk.checked)
```

```
                form.chk.checked=true;                     //将复选框置于选中状态
        }else{                                             //undefined，说明有多个复选框
          for( var i = 0; i < leng; i++){                  //使用 for 循环，将所有复选框选中
                if(!form.chk[i].checked)
                        form.chk[i].checked = true;
            }
        }
        return false;
}
/*    反选，就是将选中的复选框取消选中，而未被选中的复选框则被选中 */
function overdel(form){
        var leng = form.chk.length;                        //获取复选框个数
        if(leng==undefined){                               //如果 leng 为 undefined，说明只有一个复选框
          if(!form.chk.checked)
                form.chk.checked=true;
            else
                form.chk.checked=false;
        }else{                                             //否则，说明有多个复选框
          for( var i = 0; i < leng; i++){                  //使用 for 循环，对所有复选框进行操作
            if(!form.chk[i].checked)                       //根据复选框 checked 属性的情况，进行反向选择
                        form.chk[i].checked = true;
                else
                        form.chk[i].checked = false;
            }
        }
        return false;
}
```

使用 alldel 或 overdel 选中复选框后，即可调用 alldel()函数来实现删除功能。alldel()函数首先使用 for 循环，将被选中的复选框的 value 值取出并存成数组；然后根据数组生成 url，并使用 xmlhttp 对象调用这个 url，当处理完毕后，根据返回值，或弹出提示，或刷新本页。该函数代码如下：

```
/*    删除记录    */
function del(form){
        if(!window.confirm('是否要删除数据??')){
        }else{
            var leng = form.chk.length;                    //返回复选框的长度
            if(leng==undefined){
                if(!form.chk.checked){
                        alert('请选取要删除数据!');
                }else{                                     //如果只有一个复选框，且处于选中状态
                        rd = form.chk.value;               //将复选框的 value 值直接赋给变量 rd
                        var url = 'delshop.php?rd='+rd;    //根据 rd 生成 url
                        xmlhttp.open("GET",url,true);      //调用 xmlhttp 对象
                        xmlhttp.onreadystatechange = delnow;  //调用 delnow 函数
                        xmlhttp.send(null);
                }
            }else{                                         //如果复选框为多个
                var rd=new Array();                        //声明数组
```

```
                var j = 0;
                for( var i = 0; i < leng; i++){                          //循环检查复选框状态
                        if(form.chk[i].checked){
                                rd[j++] = form.chk[i].value;             //将被选中的复选框的 value 值存到 rd 内
                        }
                }
                if(rd == ''){
                        alert('请选取要删除数据!');
                }else{
                        var url = "delshop.php?rd="+rd;                  //解释同上
                        xmlhttp.open("GET",url,true);
                        xmlhttp.onreadystatechange = delnow;
                        xmlhttp.send(null);
                }
        }
    }
    return false;
}
/*      显示状态   */
function delnow(){
    if(xmlhttp.readyState == 4){
        if(xmlhttp.status == 200){
                var msg = xmlhttp.responseText;                          //获取 xmlhttp 对象返回的文本值
                if(msg != '1'){
                        alert('删除失败');                               //如果为 1，说明删除失败
                }else{
                        alert('删除成功');
                        location=('?page=shopcar');                      //否则，说明删除成功，刷新购物车
                }
        }
    }
}
```

26.12.5　保存购物车

当用户希望保存商品数量时，可以单击“继续购物”按钮。这时，系统将触发 onclick 事件调用 conshop()函数来保存数据。该函数有一个参数，就是当前表单的名称。在 conshop()函数内，根据复选框和商品数量文本域，生成两个数组 fst 和 snd，分别保存商品 ID 和商品数量。这里要注意：两个数组的值是要相互对应的，如商品 1 的 ID 保存到 fst[1]中，那么商品 1 的数量就要保存到 snd[1]中。然后根据这两个数组生成一个 url，使用 XMLHttpRequest 对象调用 url，最后根据回传信息作出相应的判断。conshop()函数的代码如下：

```
/*      更改商品数量     */
function conshop(form){
    var n_pre = 'cnum';                                                  //商品数量文本域前缀
    var lang = form.chk.length;                                          //获取复选框的数量
/*      如果只有一个复选框，那么将商品 ID 和商品数量直接保存到变量中   */
```

```
        if(lang == undefined){
                var fst = form.chk.value;
                var snd = form.cnum0.value;
/*      否则，将商品 ID 和对应的商品数量保存到两个相应的数组中      */
        }else{
                var fst= new Array();                            //商品 ID 数组
                var snd = new Array();                           //商品数量数组
/*      循环获取复选框的 value 值和商品数量文本框的 value 值    */
                for(var i = 0; i < lang; i++){
                        var nm = n_pre+i.toString();
                        var stmp = document.getElementById(nm).value;
/*      对商品数量文本框进行判断，不允许为空，而且不允许为非数字输入 */
                        if(stmp    == '' || isNaN(stmp)){
                                alert('不允许为空、必须为数字');
                                document.getElementById(nm).select();
                                return false;
                        }
                        snd[i] = stmp;
                        var ftmp = form.chk[i].value;
                        fst[i] = ftmp;
                }
        }
        var url = 'changecar.php?fst='+fst+'&snd='+snd;          //生成 url
        xmlhttp.open("GET",url,true);                            //调用 xmlhttp 对象
        xmlhttp.onreadystatechange = updatecar;                  //调用 updatecar 函数
        xmlhttp.send(null);
}
/*      对 xmlhttp 对象的返回值进行处理      */
function updatecar(){
        if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                if(msg == '1'){
                        location='index.php';                    //如果操作成功，返回首页
                }else{
                        alert('操作失败');                       //否则提示错误
                }
        }
}
```

函数中调用的 changecar.php 页为数据处理页，该页将商品 ID 和商品数量进行重新排列，并保存到 shopping 字段内。该页面代码如下：

```
<?php
        session_start();                                         //开启 Session 支持
        include_once 'conn/conn.php';                            //载入数据库连接文件
        $sql = 'select id,shopping from tb_user';                //生成 SQL 查询语句
        $sql .= ' where id = '.(int)$_SESSION['id'];
        $fst = $_GET['fst'];                                     //获取商品 ID
        $snd = $_GET['snd'];                                     //获取商品数量
        $reback = '0';                                           //设置返回值，默认为 0
```

26.13.1 显示订单

formset()函数的作用是将商品信息整理后，通过 open 方法打开 settle.php 页来显示订单，并将整理后的商品信息一并传过去。formset()函数的代码如下：

```
function formset(form){
var uid = form.uid.value;                               //获取订单提交人的名称
var n_pre = 'cnum';                                     //商品数量文本前缀
    var lang = form.chk.length;                         //复选框个数
    /*    如果复选框只有 1 个 */
    if(lang == undefined){
        var fst = form.chk.value;                       //商品 ID
        var snd = form.cnum0.value;                     //商品数量
    /*    当有多个复选框时    */
    }else{
        var fst= new Array();
        var snd = new Array();
    /*    使用 for 循环，获取商品数量和商品 ID    */
        for(var i = 0; i < lang; i++){
            var nm = n_pre+i.toString();
            var stmp = document.getElementById(nm).value;
            if(stmp    == '' || isNaN(stmp)){
                alert('不允许为空、必须为数字');
                document.getElementById(nm).select();
                return false;
            }
            snd[i] = stmp;
            var ftmp = form.chk[i].value;
            fst[i] = ftmp;
        }

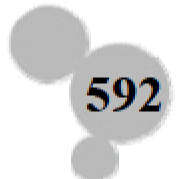
    }
    /*    使用 open 方法，打开 settle.php 页，并将当前会员名称、商品 ID 和商品数量的值传过去  */
    open('settle.php?uid='+uid+'&fst='+fst+'&snd='+snd,'_blank','width=420 height=220',false);
}
```

说明：因为 open 方法使用了_blank 参数来打开一个新的页面，Session 值传不过去。所以这里使用隐藏域来传递会员名称。

26.13.2 填写订单

settle.php 直接将接收的值传给 settle.tpl 模板，并载入 settle.tpl 模板。settle.php 页面的代码如下：

```
<?php
    include_once 'config.php';                      //载入 Smarty 配置文件
    $fst = $_GET['fst'];                            //获取 get 属性值
```



settle_chk.php 页的代码如下：

```
<?php
/*	获取表单值，并存储到添加数组中	*/
	include_once 'conn/conn.php';
	$sql = "select * from tb_form where id = -1";
	$rst = $conn->execute($sql);
	$addform = array();
	$addform['vendee'] = $_POST['uid'];				//提交订单用户
	$addform['commo_id'] = $_POST['fst'];				//购买商品 ID
	$addform['commo_num'] = $_POST['snd'];				//购买商品数量
/*	根据商品 ID 和商品数量，获取所需数据	*/
	$addform['formid'] = time()					//订单号
	$tmpid = explode(',',$addform['commo_id']);			//将商品 ID 存为数组
	$tmpnm = explode(',',$addform['commo_num']);			//将商品数量存为数组
	$number = count($tmpid);					//获取商品数组长度
	/*	当购买商品种类大于 1 时	*/
	if($number >1){
		$tmpna = array();					//商品名称数组
		$tmpvp = array();					//商品单价数组
		$tmpfd = array();					//商品折扣数组
		$tmptt = 0;						//商品总额数组
		/*	使用 for 循环获取各种商品的信息	*/
		for($i = 0; $i < $number; $i++){
			/*	生成查询语句，执行 SQL，最后返回查询结果集	*/
			$tmpsql = "select name,v_price,fold from tb_commo where id = ".$tmpid[$i];
			$tmprst = $conn->execute($tmpsql);
			/*	将每种商品的信息，放到对应的数组中	*/
			$tmpna[$i] = $tmprst->fields['name'];
			$tmpvp[$i] = $tmprst->fields['v_price'];
			$tmpfd[$i] = $tmprst->fields['fold'];
			$tmptt += $tmprst->fields['v_price'] * $tmpnm[$i];
		}
		/*	将获取的信息数组分别生成长字符串，数组值之间使用逗号“,”分隔	*/
		$addform['commo_name'] = implode(',',$tmpna);		//商品名称
		$addform['agoprice'] = implode(',',$tmpvp);		//商品价格
		$addform['fold'] = implode(',',$tmpfd);			//商品折扣
		$addform['total'] = $tmptt;				//商品总额
	/*	当购买一种商品时	*/
	}else if($number == 1){
		$tmpsql = "select name,v_price,fold from tb_commo where id = ".$tmpid[0];
		$tmprst = $conn->execute($tmpsql);
		$addform['commo_name'] = $tmprst->fields['name'];
		$addform['agoprice'] = $tmprst->fields['v_price'];
		$addform['fold'] = $tmprst->fields['fold'];
		$addform['total'] = $tmprst->fields['v_price'] * $tmpnm[0];
	/*	没有商品时	*/
	}else{
		echo 'error';
```

```
			exit();
		}
		/*	将表单的其他信息存入添加数组		*/
		$addform['taker'] = $_POST['taker'];					//收货人
		$addform['code'] = $_POST['code'];					//邮编
		$addform['tel'] = $_POST['tel'];					//联系电话
		$addform['address'] = $_POST['address'];				//地址
		$addform['del_method'] = $_POST['del'];				//送货方式
		$addform['pay_method'] = $_POST['pay'];				//付款方式
		$addform['state']	= '0';						//订单状态，0 为未处理
		$InsertSQL = $conn->GetInsertSQL($rst,$addform);			//生成 SQL 添加语句
		if(false == $conn->execute($InsertSQL)){
			/*	如果购买失败，返回上层	*/
			echo "<script>alert('购买失败');history.back;</script>";
		}else{
			/*	如果成功，删除购物车信息，并显示订单信息	*/
			$updsql = "select * from tb_user where name = '".$_POST['uid']."'";
			$updrst = $conn->execute($updsql);
			$arr = array();
			$arr['consume'] = $addform['total'];
			$arr['shopping'] = '';
			$UpdateSQL = $conn->GetUpdateSQL($updrst,$arr);
			$conn->execute($UpdateSQL);
			echo "<script>top.opener.location.reload();</script>";
			echo "<script>open('forminfo.php?fid=$fid','_blank','width=600 height=450',false);</script>";
			echo "<script>window.close(); </script>";
		}
?>
```

26.13.4　反馈订单

订单提交成功后，forminfo.php 页会将新添加的订单反馈给用户，并提醒用户保存订单号，以便根据订单号来付款、查询及处理问题。反馈订单页面的运行结果如图 26.26 所示。

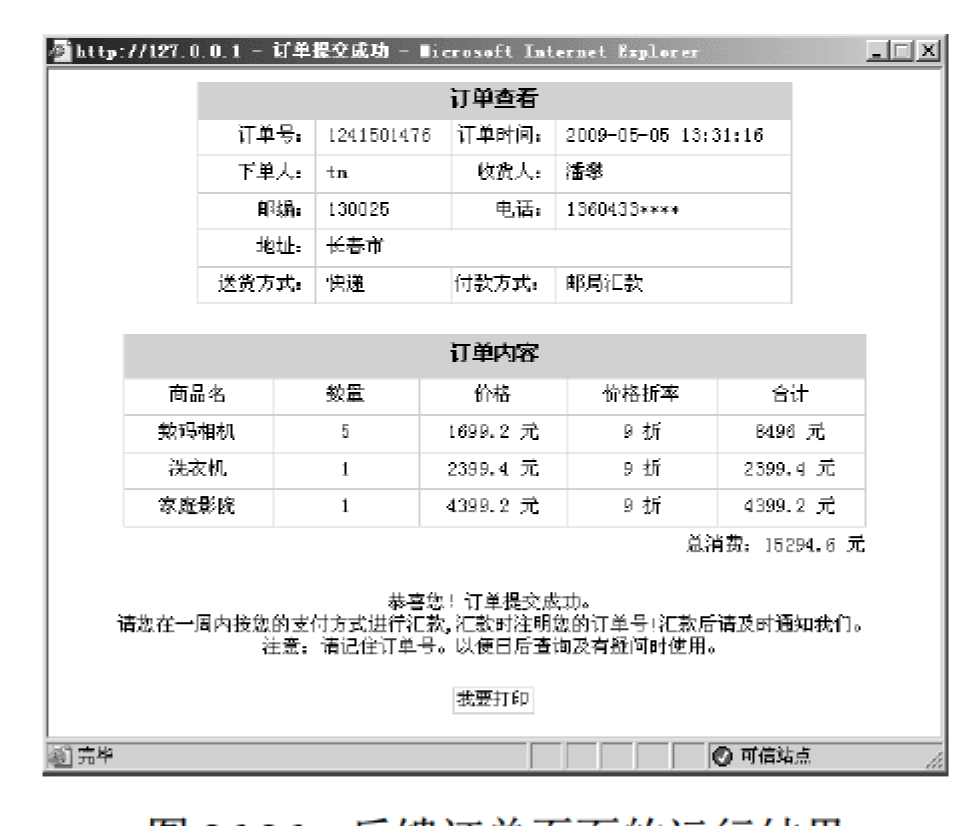


图 26.26　反馈订单页面的运行结果

forminfo.php 页的代码如下：

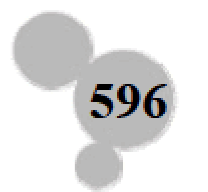
```
<?php
    include_once 'conn/conn.php';                          //载入数据库连接文件
    include_once 'config.php';                             //载入 Smarty 模板文件
    $id = $_GET['fid'];                                    //获取订单 ID
    $sql = "select * from tb_form where id = ".$id;       //根据 ID 生成查询语句
    $rst = $conn->execute($sql);                           //执行 SQL 语句，返回结果集
    $formarr=$rst->GetArray();                             //将结果集存储为数组
    $commname = explode(',',$formarr[0]['commo_name']);    //将商品名称存储为数组
    $commnumber = explode(',',$formarr[0]['commo_num']);   //将商品数量存储为数组
    $commagoprice = explode(',',$formarr[0]['agoprice']);  //将商品单价存储为数组
    $commfold = explode(',',$formarr[0]['fold']);          //将商品折扣率存储为数组
    /*    将各个数组传给模板，最后显示模板        */
    $smarty->assign('formarr',$formarr[0]);
    $smarty->assign('commname',$commname);
    $smarty->assign('commnumber',$commnumber);
    $smarty->assign('commagoprice',$commagoprice);
    $smarty->assign('commfold',$commfold);
    $smarty->assign('title','订单提交成功');
    $smarty->display('forminfo.tpl');
?>
```

模板页 forminfo.tpl 包含了两个表格，第一个表格显示提交订单用户所填信息；第二个表格为购买的商品信息。最后还有一行注意文字，提醒用户不要忘记订单号。如果有条件，还可以将订单打印出来。模板页商品信息代码如下：

```
<!--   商品信息    -->
<table border="0" align="center" cellpadding="0" cellspacing="0">
  <tr><td colspan="5">订单内容</td></tr>
  <tr><td>商品名</td><td>数量</td><td>价格</td><td >价格折率</td><td>合计</td></tr>
{foreach key=key item=item from=$commname}
<tr>
    <td>{$item}</td><td>{$commnumber[$key]}</td>
    <td>{$commagoprice[$key]} 元</td><td>{$commfold[$key]} 折</td>
    <td>{$commagoprice[$key]*$commnumber[$key]} 元</td>
</tr>
{/foreach}
<tr><td colspan="5">总消费：{$formarr.total} 元</td></tr></table>
<p  align="center">恭喜您！订单提交成功。<br />请您在一周内按您的支付方式进行汇款,汇款时注明您的<font color="red">订单号</font>!汇款后请及时通知我们</p>
<p ><font color="red">注意：</font>请记住<font color="red">订单号</font>。以便日后查询及有疑问时使用。</p>
<p align="center"><input type="button" value="我要打印" onclick="window.print()" class="btn" /> </p>
```

26.13.5　查询订单

订单提交后，可以通过导航栏中的“订单查询”超链接来查看订单情况。查询关键字可以是会员



名称或订单号，只要输入一项就可以。单击“查询”按钮，将会在页面下方显示查询结果。查询订单页面运行结果如图 26.27 所示。

查询订单

查询用户：	tm	查询订单号：		查询

查询结果

订单号	订货会员	收货人	订单金额	付款方式	收款方式	订单状态
1209117958	tm	sp	6666	银行转账	平邮	已付款
1209119797	tm	spcn	73326	银行转账	平邮	未作处理
1241501476	tm	潘攀	15294.6	邮局汇款	快递	未作处理

图 26.27　查询订单页面运行结果

本模块采用 Ajax 技术，对查询结果在本页进行直接输出。共使用 4 个文件：queryform.php、queryform.tpl、queryform.js 和 query.php。

queryform.php 页用于载入 Smarty 配置文件，并通过 display 方法指定 queryform.tpl 模板页。

queryform.tpl 包含了一个查询表单和一个 div 标签，其元素属性和名称如表 26.3 所示。

表 26.3　查询表单元素和 div 标签属性及说明

元 素 名 称	元 素 属 性	元 素 说 明
<form>	id='queryform' name='queryform' method='post' action='#'	查询表单，名称为 queryform
text	input id='name' name='name' type='text' class='txt'	会员查询文本框
text	id='formid' name='formid' type='text' class='txt'	订单号查询文本框
button	id='enter' name='enter' type='button' value='查询' class='btn' onClick='return queryrst(queryform)'	查询按钮，调用 queryrst 函数
<div>	id=exam	用于显示查询结果

通过表 26.3 可以看到，当单击“查询”按钮时，将触发 onclick 事件，调用 queryrst()函数。queryrst()函数在脚本文件 js/queryform.js 内。该文件代码如下：

```
function queryrst(form) {
	/*	获取会员名称	*/
	var name = form.name.value;
	/*	获取订单号	*/
	var formid = form.formid.value;
	/*	如果两者都为空，则弹出警告框	*/
	if(name == '' && formid == ''){
		alert('用户或订单号至少有一个不能为空');
		form.name.focus();
		return false;
	}
	/*	将获取的变量传给 query.php 页*/
	var url = "query.php?vendee="+name+"&formid="+formid;
	xmlhttp.open("GET",url,true);
	/*	调用 showfm()函数	*/
	xmlhttp.onreadystatechange = showfm;
	xmlhttp.send(null);
}
function showfm(){
```

```
                    case 2:$tmp .= '已发货';break;
                    case 3:$tmp .= '已收货';break;
                }
                $tmp .= "</td></tr>";
            }
            $tmp .= "</table>";
        /*  将临时变量赋给$reback 并输出        */
            $reback = $tmp;
        }
        echo $reback;
?>
```

有关电子商务平台前台功能模块的讲解到此结束，接下来讲解后台管理系统中首页、类别管理和订单管理模块的设计。

26.14　后台首页设计

视频讲解：光盘\TM\Video\26\后台首页设计.exe

后台管理系统对网站的商品、商品类别、会员及公告等信息进行统一管理。本系统的后台主要包括如下功能模块。

- ☑ 类别管理模块：主要包括对商品类别的添加、修改及删除操作。
- ☑ 商品管理模块：主要包括对商品的添加、修改、删除及订单处理。
- ☑ 用户管理模块：主要包括管理员管理和会员管理。其中，管理员管理是实现对管理员的添加、删除和修改功能；会员管理则包括删除和冻结功能。
- ☑ 公告管理模块：主要包括公告的添加及删除操作。
- ☑ 链接管理模块：主要包括添加、修改和删除友情链接。

后台首页的运行结果如图 26.28 所示。



图 26.28　后台首页运行结果

26.14.1　后台首页布局

后台首页和前台首页不同，使用的是框架布局。main.php 页的框架代码如下：

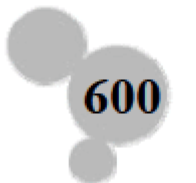
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<link rel="stylesheet" href="css/style.css" />
<title>明日购物商城后台管理系统</title>
</head>
<!--  使用 frame 框架布局  -->
<frameset rows="113,*,100" cols="1004" frameborder="no" border="1" framespacing="0">
    <!--  头部框架，命名为 topFrame，载入文件为 top.php  -->
    <frame src="top.php" name="topFrame" scrolling="No" noresize="noresize" id="topFrame" title="topFrame" />
    <frameset rows="*" cols="310,*" framespacing="0" frameborder="no" border="1">
        <!--  左侧框架，名字为 leftFrame，载入文件为 left.php  -->
        <frame src="left.php" name="leftFrame" frameborder="0" scrolling="auto" noresize="noresize"
id="leftFrame" title="leftFrame" />
        <1--  中部框架，名字为 mainFrame，载入文件是 default.php  -->
        <frame src="default.php" name="mainFrame" id="mainFrame" title="mainFrame" />
    </frameset>
    <1--  底部框架，名字为 bottomFrame，载入文件是 bottom.php  -->
    <frame src="bottom.php" name="bottomFrame" scrolling="No" noresize="noresize" id="bottomFrame"
title="bottomFrame" />
</frameset>
<noframes><body></body></noframes>
</html>
```

在框架页中包含了 4 个文件：top.php 为后台页面的标题；left.php 为导航栏；default.php 为后台的功能输出页面；bottom.php 为网站的尾文件。

26.14.2　DIV+JavaScript+CSS 实现树状菜单

在 left.php 页面中设计了一个树状菜单，是应用 DIV+JavaScript+CSS 来实现的。首先来看 div 标签。

```
<!--  载入 css 样式和 javascript 脚本  -->
<link href="css/left.css" rel="stylesheet" type="text/css" />
<script language="javascript" src="js/left.js"></script>
<!--  类别管理菜单，注意加粗的地方  -->
<div id="type" align="center" onclick="javascript:change(one,type);">类别管理</div>
<!--  子菜单  -->
<div id="one" style="display: ">
<div id="addtype" align="center"><a href="addtype.php" target="mainFrame" id="menu">添加类别</a></div>
<div id="showtype" align="center"><a href="showtype.php" target="mainFrame" id="menu">查看类别</a></div>
</div>
<div id="hidediv" align="center"></div>
<!--  商品管理菜单  -->
<div id="commo" align="center" onclick="javascript:change(two,type);">类别管理</div>
<div id="two" style="display:none">
<!--  商品管理子菜单  -->
    …
</div>
```



加商品的类别。当选择“一级类别”时，下面的“父类名称”则不可选。添加类别页面的运行结果如图 26.30 所示。

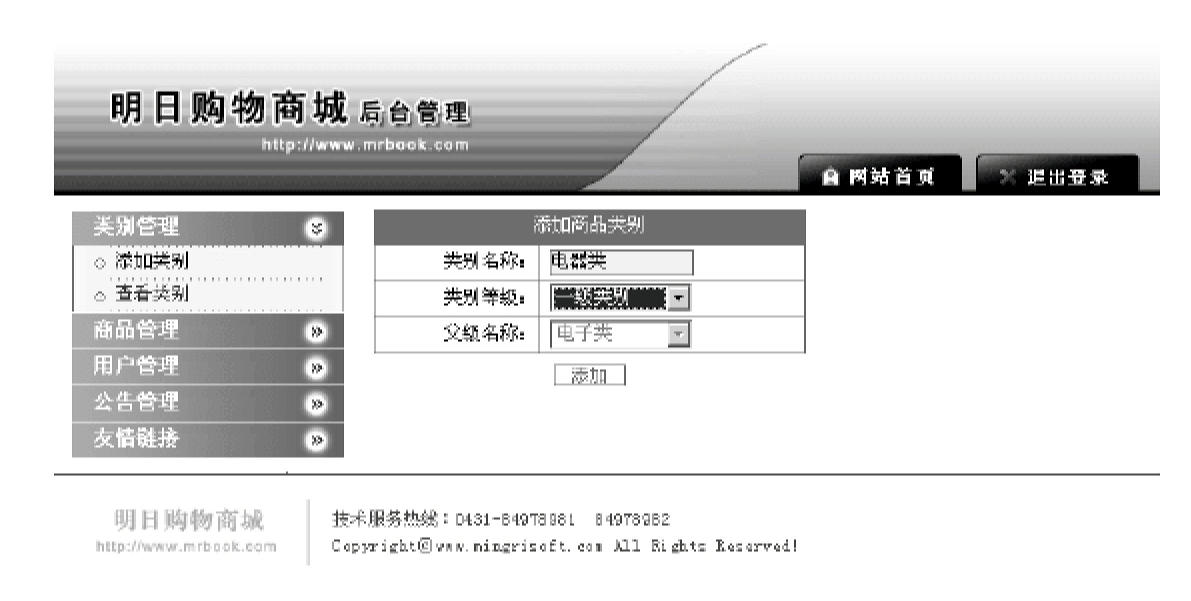


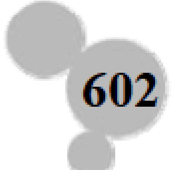
图 26.30　添加类别页面的运行结果

当单击“添加类别”超链接时，将调用 addtype.php 页。该页从数据库中读取所有 supid=0（也就是一级类别）的记录，并将结果集直接输出为下拉菜单，传递给模板页。addtype.php 页的代码如下：

```
<?php
	/*	载入数据库连接文件和 Smarty 模板配置文件	*/
	include_once 'config.php';
	include_once 'conn/conn.php';
	/*	生成查询语句，查询所有 supid=0 的一级类别	*/
	$sql = "select name,id from tb_class where supid = 0";
	/*	执行 SQL 语句，返回结果集	*/
	$rst = $conn->execute($sql);
	/*	将结果集直接输出为下拉菜单，并传递给模板	*/
	$smarty->assign('op',$rst->GetMenu2("supid",'',$blank = false,'','','class="txt"'));
	$smarty->display('addtype.tpl');
?>
```

模板页 addtype.tpl 接受 op 下拉菜单后，将其显示在“添加商品类别”表单中。addtype.tpl 模板代码如下：

```
<!--	载入 js 脚本文件	-->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/chktype.js"></script>
<table width="300" border="0" align="center" cellpadding="0" cellspacing="0">
<form id="addtype" name="addtype" method="post" action="#">
	<tr><td colspan="2">添加商品类别</td></tr>
	<tr><td >类别名称：</td><td><input name="names" type="text" id="names" class="txt"></td></tr>
	<tr><td >类别等级：</td><td>
	<!--	类别等级下拉菜单，会触发 onchange 事件，调用 changetype()函数	-->
			<select name="grade" OnChange="changetype(addtype)" >
				<option value="1">一级类别</option>
				<option value="2" selected>二级类别</option>
			</select>
		</td></tr>
	<!--	显示 op 变量	-->
	<tr><td>父级名称：</td> <td>{$op}</td></tr>
	<tr><td colspan="2"><input type="button" value="添加" class="btn" onClick="chktype(addtype)"></td>
```



```
    </tr>
</form>
</table>
```

该模板通过“类别等级”下拉菜单的 changetype()函数来控制 op 菜单的状态。如果添加的是一级类别，则 op 菜单不可用；如果添加的是二级菜单，则 op 菜单可用。当用户输入添加类别后，单击“添加”按钮时，会调用 chktype()函数。包含这两个函数的 chktype.js 脚本文件的代码如下：

```
/*	控制 op 菜单的状态		*/
function changetype(form){
		if(form.grade.value=="2"){
			form.supid.disabled=false;
		}else{
			form.supid.disabled=true;
}}
/*	处理“类别名称”文本框		*/
function chktype(form){
	var name = form.names.value;
	var supid = form.supid.value;
	if(form.names.value == ""){
		alert('请填写类别名称');
		form.names.focus();
		return false;
	}
/*	根据不同的类别等级，生成不同的 url		*/
	if(form.grade.value == "1"){
		var url = "chktype.php?name="+name;
	}else{
		var url = "chktype.php?name="+name+"&supid="+supid;
	}
/*	使用 xmlhttp 对象打开该 url，并调用 check 函数处理返回值		*/
	xmlhttp.open("GET",url,true);
	xmlhttp.onreadystatechange = check;
	xmlhttp.send(null);
}
/*	根据返回值作出相应的处理		*/
function check(){
	if(xmlhttp.readyState == 4){
		if(xmlhttp.status == 200){
			var msg = xmlhttp.responseText;
			if(msg == "1"){
				alert('类名重复');
			}else if(msg == "2"){
				alert('添加失败!');
			}else{
				alert('添加成功');
				location='showtype.php';
}}}}
```

在 chktype.php 处理页实现对类别的添加，并返回不同的参数以便 chktype()函数进行处理。

chktype.php 页的代码如下：

```
<?php
	include_once 'conn/conn.php';                              //载入数据库连接文件
	$name = $_GET['name'];                                     //获取类别名称
	($_GET['supid'] == '')?($supid = 0):($supid = $_GET['supid']);   //获取类别等级
	$reback = '';                                              //设置返回变量
	$sql = "select * from tb_class where name = '$name'";      //声明 SQL 语句
	$rst = $conn->execute($sql);                               //返回查询结果集
	if($rst->RecordCount() == 1){                              //如果有查询结果，说明类别名被占用
		$reback = '1';
	}else{                                                     //否则执行添加操作
		$ine = array();
		$ine["name"] = $name;
		$ine["supid"] = $supid;
		$intsql = $conn->GetInsertSQL($rst,$ine);              //生成添加语句
		if($conn->execute($intsql) == false){
			$reback = '2';
		}else{
			$reback = '3';                                     //如果成功，返回 3
		}}
	echo $reback;
?>
```

26.15.2　查看类别

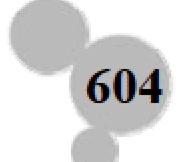
类别添加成功后，可以通过“查看类别”超链接来修改、删除类别。在查看商品页中，类别按照父类 1：子类 1、子类 2……；父类 2：子类 1、子类 2……的格式显示。修改类别名称时，直接在对应的类别文本框中输入新类名，然后单击对应的“修改”按钮即可。删除时则直接单击对应的“删除”按钮。当要删除的父类包含子类时，则不允许删除父类。查看类别页面的运行结果如图 26.31 所示。



图 26.31　查看类别页面运行结果

首先来介绍 showtype.php 页。该页执行两次查询，分别输出一级类别和二级类别的数组，传递给模板页。showtype.php 页的代码如下：

```
<?php
	include_once 'config.php';                                 //载入 Smarty 配置文件
```



```
	include_once 'conn/conn.php';							//载入数据库连接文件
	$bigsql = 'select id,name from tb_class where supid = 0';			//生成一级类别查询语句
	$smallsql = 'select * from tb_class where supid != 0';			//生成二级类别查询语句
	$bigclass = $conn->execute($bigsql);					//返回一级类别结果集
	$smallclass = $conn->execute($smallsql);					//返回二级类别结果集
	$bigarray = $bigclass->GetAssoc();					//将结果集输出成数组类型
	$smallarray = $smallclass->GetAssoc();
	$smarty->assign('bigarray',$bigarray);					//将数组传递给模板
	$smarty->assign('smallarray',$smallarray);
	$smarty->display('showtype.tpl');
?>
```

showtype.tpl 模板页通过嵌套 foreach 循环，将一级类别和二级类别一一对应起来，并输出到页面。showtype.tpl 页的代码如下：

```
<!--	载入 js 脚本文件-->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/changetype.js"></script>
<table>
<!--	省略 html 无用代码	-->
…
<!--	外层循环，输出一级类别	-->
{foreach name=ftype key=fkey item=fitem from=$bigarray}
	<tr><td>父类：
	<input id="moditype{$fkey}" name="moditype{$fkey}" type="text" class="shorttxt" value="{$fitem}" style="border-color:#996633;" /></td>
		<td>
<input id = " modify " name = " modify " type = " button " value = "  修改  " onclick = " modifytype({$fkey}); "/>
<input id = " delete " name = " delete " type = " button " value = "删除" onclick = "delbigtype({$fkey});">
	</td></tr>
<!--	内层循环，输出对应的二级类别	-->
	{foreach name = stype key = skey item = sitem from = $smallarray}
	{if $sitem[1] == $fkey}
	<tr><td>子类：
	<input id="modtype{$skey}" name="moditype{$skey}" type="text" value="{$sitem[0]}"/></td>
		<td><input id="modidfy" name="modify" type="button" value="修改" onclick=" modifytype({$skey})"/>
		 <input id="delete" name="delete" type="button" value="删除"onclick=" delsmalltype({$skey})">
		 </td></tr>
	 {/if}
	{/foreach}
{/foreach}
</form>
</table>
```

showtype.tpl 模板中包含的 changetype.js 脚本文件主要有 3 个函数：modifytype()、delbigtype()和 delsmalltype()，分别控制类别的修改和删除。该脚本文件的代码如下：

```
/*	修改类别名称函数	*/
function modifytype(key){
	var nm = 'moditype'+key;
```

```
<!--   载入 js 脚本文件   -->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/shwform.js"></script>
<table border="0" cellspacing="0" cellpadding="0" align="center">
  <form id="shwfrm" name="shwfrm" method="post" action="#">
    <tr>
<!--   省略部分为要显示的订单信息的字段名   -->
     …
      <td>订单状态</td>
      <td>处理</td>
    </tr>
    {foreach key=key item=item from=$formarr}
    <tr>
<!--   省略部分为显示的订单信息   -->
      <td>
<!--   根据不同的状态值，显示不同的结果   -->
{if $item.state == 0}未处理{elseif $item.state == 1}已收款{elseif $item.state == 2}已发货{elseif $item.state == 3}已收
货{/if}
      </td>
<!--   单击“请求处理”按钮，调用 showme()函数   -->
      <td><input id="deal" name="deal" type="button" value="请求处理" onclick="showme({$item.formid})" />
</td>
    </tr>
    {/foreach}
  </form>
</table>
<!--   隐藏的表单，ID 为 chdl   -->
<div id="chdl" style="display:none;">
  <table width="525" border="0" align="center" cellpadding="0" cellspacing="0">
    <form id="changedeal" name="changedeal" method="post">
      <tr><tdcolspan="6"">订单处理</td></tr>
      <tr><td>订单号：</td>
<!--   当隐藏表单显示时，该<div>标签将显示被激活订单的 formid   -->
        <td><div id="formid"> </div></td>
<!--   下面为一组单选按钮，它们有着相同的名称和不同的 ID   */
        <td><input id="acceptmon" name="acceptsend" type="radio" value="1" checked="checked" />已收款</td>
        <td><input id="sendwa" name="acceptsend" type="radio" value="2" />已发货</td>
        <td><input id="acceptwa" name="acceptsend" type="radio" value="3" />已收货</td>
<!--   单击“修改”按钮，会调用 changeme()函数来更新订单状态   -->
        <td><input id="chg" name="chg" type="button" value="修改" onclick="return changeme(changedeal)" /></td>
      </tr>
    </form>
  </table>
</div>
```

在该页中所调用的函数都包含在 shwform.js 脚本文件中。shwome()函数的作用是改变 id=chdl 的<div>标签的状态，唯一的参数是该条订单的订单号（formid），并在 id=formid 的<div>标签中显示出来；另一个函数 changeme()的作用是将 formid 值和被选中的单选按钮值取出并传给 changestate.php 进行处理，根据处理返回值刷新本页面。shwform.js 页面的主要代码如下：

```
/*    显示/隐藏表单  */
function showme(formid){
    /*    判断“id=chdl”的<div>标签的样式，如果是显示，则改为隐藏 */
    if(document.getElementById('chdl').style.display == ''){
        document.getElementById('chdl').style.display = 'none';
    /*    否则，将该标签显示出来，并将 formid 显示在“id=formid”的<div>标签中     */
    }else{
        document.getElementById('chdl').style.display = '';
        document.getElementById('formid').innerHTML = formid;
    }
}
/*    改变状态  */
function changeme(form){
    /*    获取 formid      */
    var fid = document.getElementById('formid').innerText;
    var lang = form.acceptsend.length;
    var state;
    /*    循环查看单选按钮的状态，并取得被选中按钮的 value 值   */
    for(var i=0; i < lang; i++){
        if(form.acceptsend[i].checked == true){
            state = form.acceptsend[i].value;
        }
    }
    if(state == undefined){
        alert('请选择处理项');
        return false;
    }
    /*    将 formid 和 state 值传给 changestate.php 页      */
    var url = "changestate.php?formid="+fid+"&state="+state;
    /*    使用 XMLHttpRequest 对象调用该 url             */
    xmlhttp.open("GET",url,true);
    /*    对返回进行处理响应       */
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
            var msg = xmlhttp.responseText;
            if(msg == '1'){
                location.reload();
            }else{
                alert('修改失败'+msg);
}}}
    xmlhttp.send(null);
}
```

最后的 changestate.php 负责将改变的记录更新到数据表中。changestate.php 页的代码如下：

```
<?php
    include_once 'conn/conn.php';                    //调用数据库连接文件
    $formid = $_GET['formid'];                       //获取参数值
    $state = $_GET['state'];
    $reback = '0';                                   //声明返回值，初始化为 0
```

```
	$arr = array();                                                  //声明更新数组
	$sql = "select * from tb_form where formid = '".$formid."'";    //根据 formid 生成查询语句
	$rst = $conn->execute($sql);                                     //返回查询结果
	if($rst->RecordCount() != 1){                                    //如果没有返回值，返回 2
		$reback = '2';
	}else{                                                           //如果有返回值
		$arr['state'] = $state;                                      //将更新数据存到数组中
		$UpdateSQL = $conn->GetUpdateSQL($rst, $arr);                //生成更新语句
		if(false == $conn->execute($UpdateSQL)){                     //执行更新
			$reback = '3';
		}else{
			$reback = '1';                                           //成功返回 1
}}
	echo $reback;                                                    //输出返回值
?>
```

26.17　开发的常见问题

视频讲解：光盘\TM\Video\26\开发的常见问题.exe

在本系统开发和后期测试的过程中，开发人员遇到了各种各样的疑难问题。这里精选一些常见的、容易被忽略的问题加以讲解，希望可以为初学者提供一些帮助，在开发程序时少走一些弯路。

26.17.1　解决 Ajax 的乱码问题

问题描述：当使用 Ajax 传递数据时，要么在数据处理页中数据不能被正确处理，要么输出返回值时显示无法识别的乱码。

解决方法：这是因为 PHP 在传递数据时使用的编码默认为 UTF-8，造成了非英文字符不能正确传递的情况。解决方法如下：

在所有的 PHP 页中都输入代码：

```
Header("Content-Type:text/html;charset=gb2312");
```

这样，所有的页面即可正确显示。

26.17.2　使用 JS 脚本获取、输出标签内容

问题描述：获取、更改表单元素值和特定标签内容。

解决方法：使用 JS 脚本获取页面内容的方式主要有两种。第一种是通过表单获取表单元素的 value 值。格式为：

```
表单名称.元素名.value
```

该方式只能获取表单中的元素值，对于其他标签元素则无能为力。而第二种方式可以通过 ID 名来

获取页面中任意标签的内容：

☑ 格式 1

```
document.getElementById('id').value;
```

☑ 格式 2

```
document.getElementById('id').innerText;
```

使用第二种方式时要注意，标签的 ID 名必须存在而且唯一，否则就会出现错误。向标签内容赋值时，则使用如下的格式：

```
id.innerHTML = '要显示的内容';
```

26.17.3 使用浮动框架做关联菜单

问题描述：在添加商品时，商品类别选项是一个关联菜单。在不刷新本页的前提下实现该功能十分复杂（当然，如果使用了 ADODB 和 Ajax，那就另当别论了）。

解决方法：这里使用了一个比较灵活的方法，就是使用浮动框架技术，将一个二级管理菜单放到单独的一个网页中，然后在框架中显示。当选择不同类别时，菜单页虽然也会刷新，但整个的大表单页则不会跟着刷新。其关键代码如下：

```
<tr>
<td>商品类别：</td>
<td>
<iframe id="menu1" name="menu1" src="menu.php" width="300" height="22" frameborder="0" scrolling="no" style=" margin-top:0px; margin-left: 0px; top:0px;"></iframe>
<input id="stype" name="stype" type="hidden" value="" />
</td>
</tr>
```

加粗的代码部分为隐藏表单，保存当前被选择的类别 ID。添加商品处理页就是从这个隐藏表单中获取商品类别的。

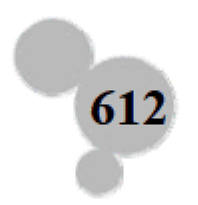
26.17.4 禁用页面缓存

问题描述：使用 Ajax 技术可以防止页面刷新，但有时也会产生新的问题。例如，在“会员管理”页面中，如果连续“冻结”和“解冻”会员，超过 3 次后，该功能就会失效。因为在一定时间内，如果做相同的操作，那么 XMLHttpRequest 对象会执行缓存里有的页面，从而造成操作失败。

解决办法：使用 header()函数将缓存关闭。使用如下语句：

```
header("CACHE-CONTROL:NO-CACHE");
```

将这句代码添加到 XMLHttpRequest 对象所调用的处理页的顶部即可。



自定义函数判断上传文件的后缀。

解决方法：创建自定义函数。f_postfix()函数的代码如下：

```
/*
 *判断文件后缀
 *$f_type：允许文件的后缀类型（数组）
 *$f_upfiles：上传文件名
 */
function f_postfix($f_type,$f_upfiles){
	$is_pass = false;
	$tmp_upfiles = split("\.",$f_upfiles);                          //使用 split()函数分割文件
	$tmp_num = count($tmp_upfiles);                                 //查找文件后缀
	if(in_array(strtolower($tmp_upfiles[$tmp_num - 1]),$f_type))    //判断后缀是否在允许列表内
		$is_pass = $tmp_upfiles[$tmp_num - 1];                  //如果是，则将后缀名赋给变量
	return $is_pass;                                                //返回变量
}
```

26.18 网站运行错误分析

视频讲解：光盘\TM\Video\26\网站运行错误分析.exe

电子商务平台开发完成，测试成功后即可投入使用，但是将程序上传到服务器、或者复制到其他机器中运行时，却又出现了一些错误。下面将分析在这种情况下出现错误的原因以及解决方法。

26.18.1 数据库导致的错误

在运行本系统时，由于没有附加数据库或者数据库的用户名和密码不正确，会出现如图 26.33 和图 26.34 所示的错误信息。



图 26.33 没有附加 MySQL 数据库导致的错误

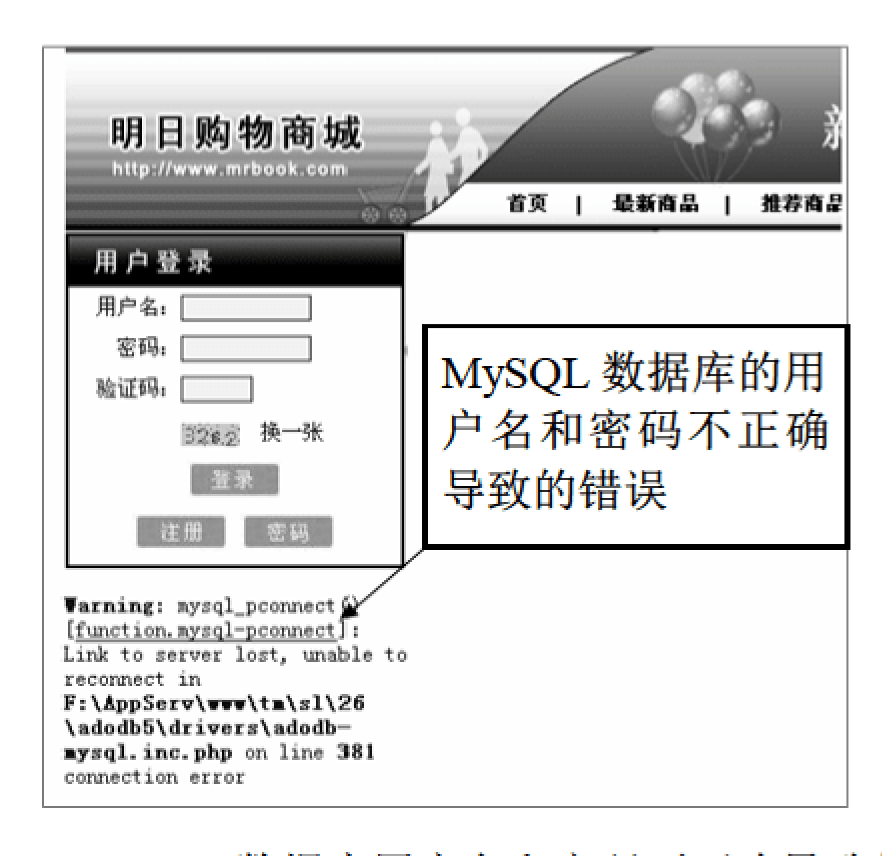
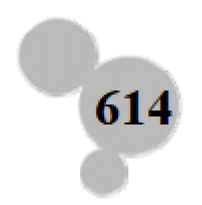


图 26.34 MySQL 数据库用户名和密码不正确导致的错误



解决方法如下：

（1）从本系统中的 26/data 文件夹下复制本系统的数据库，附加到本机 MySQL 数据库的 data 文件夹下。

（2）如果 MySQL 数据库的用户名和密码与本系统所使用的用户名和密码不一致，有如下两种解决方案。

☑ 第一种，修改本系统的数据库连接文件，找到 conn 文件夹下的 conn.php 文件，修改其中的用户名和密码，使其与本机 MySQL 数据库的用户名和密码一致。conn.php 文件的代码如下：

```
<?php
    include_once ('inc/char.php');
    include_once ("adodb5/adodb.inc.php");
    $conn = ADONewConnection('mysql');
    $conn->PConnect('localhost','root','root','db_database26') or die('connection error');
    $conn->Execute('set names gb2312');
    $ADODB_FETCH_MODE = ADODB_FETCH_BOTH;
?>
```

本系统中 MySQL 数据库的用户名为 root，密码为 root，数据库名称为 db_database26。

注意： 如果要修改本系统连接 MySQL 数据库的用户名和密码，需要修改两个 conn.php 文件，一个在 26/conn 文件夹下，另一个在 26/admin/conn 文件夹下。

☑ 第二种，修改本机 MySQL 数据库的用户名和密码，该操作可以通过 phpMyAdmin 图形化管理工具来完成，具体操作方法可以参考本书第 18 章的实战。

26.18.2　Smarty 模板的配置错误

由于本系统是应用 Smarty 模板进行开发的，所以在运行本系统时需要对 Smarty 模板进行配置，如果没有正确指定 Smarty 类库的位置或者 Smarty 类库不存在，将出现如图 26.35 所示的错误。



图 26.35　配置 Smarty 的路径不正确

上述错误出现在本系统的 config.php 文件的第 7 行，即在指定的路径下没有找到 Smarty.class.php 文件。

修改这个错误之前，首先要明确几个目录在用户机器上的位置：

☑ 确定用户计算机上 Apache 服务器的根目录（即 PHP 程序运行文件夹）。

☑ 确定本系统在 Apache 服务器根目录下的位置。

☑ 确定 Smarty.class.php 文件在本系统中的位置。

在明确了这 3 个目录之后，再来修改这个错误信息：

☑ 确定 Apache 服务器根目录“F:\AppServ\www”。

☑ 确定本系统在服务器下的位置“\tm\sl\”。

☑ 确定 Smarty.class.php 文件在本系统中的位置“26\Smarty\Smarty.class.php”。

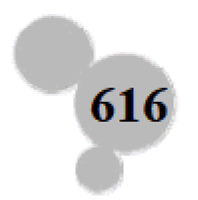
确定了上述 3 个目录以后，该错误即修改完成。在“F:\AppServ\www\tm\sl\26”文件夹下打开 congfig.php 文件，锁定第 7 行的内容，其中常量 BASE_PATH 指定的是服务器的根目录，没有任何问题；错误出现在常量 SMARTY_PATH 指定的 Smarty 类库文件的绝对路径上，图 26.35 所示指定 Smarty 类库文件的绝对路径是“\sl\26\Smarty\\”，而实际的绝对路径应该是“\tm\sl\26\Smarty\\”。config.php 的正确代码如下：

```
<?php
/*  定义服务器的绝对路径  */
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);          //获取服务器的根目录 D:/AppServ/www
/*  定义 Smarty 目录的绝对路径  */
define('SMARTY_PATH','\tm\sl\26\Smarty\\');     //指定 Smarty.class.php 文件在这个服务器中的实际位置
/*  加载 Smarty 类库文件     */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/*  实例化一个 Smarty 对象  */
$smarty = new Smarty;
/*  定义各个目录的路径     */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs/';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';
?>
```

注意： *在本系统中有两个 config.php 文件，一个直接存储在 26/文件夹下，另一个是后台管理系统的 config.php 文件，存储在 26/admin/文件夹下。如果要修改 Smarty 类库文件的绝对路径，这两个文件都要修改。*

26.19 本章小结

本章使用 Smarty、ADODB、Ajax 等目前的主流技术实现了一个电子商务平台。其中，应用 Smarty 模板实现网站的动静分离，通过 ADODB 操作 MySQL 数据库，通过 Ajax 实现无刷新验证，通过 DIV、JavaScript 和 CSS 实现树状导航菜单，并且对在开发过程中出现的疑难问题进行了总结。希望读者能通过这个项目实例，将本书前面章节中讲解的内容融会贯通，掌握网络开发的整体操作流程。



第27章

办公自动化系统

办公自动化英文原称 Office Automation，缩写为 OA，办公自动化系统即能实现办公室内事务性业务的自动化。

办公自动化没有明确的定义，最普遍的说法是：凡是在传统的办公室中采用各种新技术、新机器、新设备从事办公业务，都属于办公自动化的领域。

办公自动化系统与办公室自动化在概念上存在一定的差别。办公自动化通常指办公室中配备具有自动化功能的设备，这些设备能实现某些办公活动自动化或某个单位业务的自动化处理；而办公自动化系统则是在办公室自动化功能的基础上发展起来，以办公自动化技术为主体，同人、组织、制度、环境等相结合的完整的系统。

通过阅读本章，可以学习到：

- 办公自动化管理系统的开发流程
- 进一步学习如何做项目需求分析与系统设计
- 页面布局中的框架布局
- 使用递归函数做多级下拉菜单
- 系统日志的实现

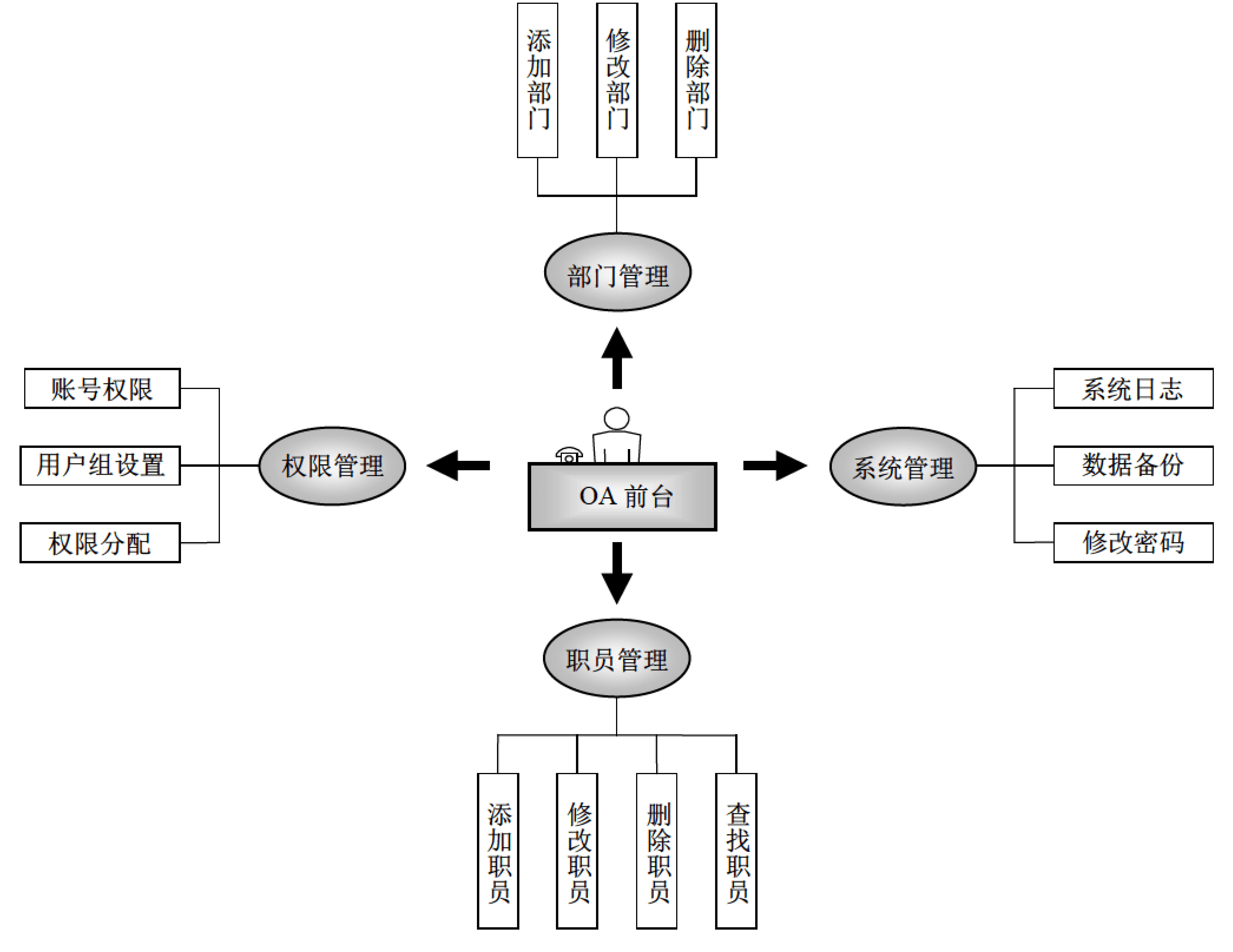


图 27.2　办公自动化后台管理系统

27.2　技 术 准 备

在开发办公自动化管理系统平台时，该项目所使用的软件及开发环境如下。

☑　服务器端

- ➢　操作系统：Windows 2003 Server/Linux（推荐）。
- ➢　服务器：Apache 2.2。
- ➢　PHP 软件：PHP 5.1.6。
- ➢　数据库：MySQL 5.0.24。
- ➢　MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- ➢　开发工具：Dreamweaver 8。
- ➢　浏览器：IE 6.0 及以上版本。
- ➢　分辨率：最佳效果 1024×768 像素。

☑　客户端

- ➢　浏览器：IE 6.0 及以上版本。
- ➢　分辨率：最佳效果 1024×768 像素。

27.3 开 发 过 程

27.3.1 数据库设计

本系统使用 MySQL 数据库，用户名为 root，密码为 root，数据库名称为 db_database27。本数据库主要包含 13 个数据表，如图 27.3 所示。



服务器: localhost ▸ 数据库: db_database27

表	类型	整理	说明
tb_company	MyISAM	gb2312_chinese_ci	企业信息列表
tb_controller	MyISAM	gb2312_chinese_ci	管理员列表
tb_depart	MyISAM	gb2312_chinese_ci	部门管理列表
tb_group	MyISAM	gb2312_chinese_ci	用户组管理列表
tb_iss	MyISAM	gb2312_chinese_ci	审核列表
tb_list	MyISAM	gb2312_chinese_ci	功能列表
tb_lyb	MyISAM	gb2312_chinese_ci	意见箱列表
tb_person	MyISAM	gb2312_chinese_ci	人事列表
tb_plan	MyISAM	gb2312_chinese_ci	计划列表
tb_register	MyISAM	gb2312_chinese_ci	登记列表
tb_setup	MyISAM	gb2312_chinese_ci	时间列表
tb_superson	MyISAM	gb2312_chinese_ci	优秀员工列表
tb_users	MyISAM	gb2312_chinese_ci	用户列表

图 27.3 db_database27 数据库中的数据表

27.3.2 前台首页设计

前台主要是员工操作页面，其功能的使用权限由后台管理员进行分配。前台首页页面如图 27.4 所示，主要显示企业公告信息、活动安排信息、个人计划信息及审核批示信息，此处的“个人计划”和“审核批示”只显示当前员工的计划和批示情况，在“任务绩效”、“质量绩效”及“批示审核”页面中可对所有员工信息进行查询。



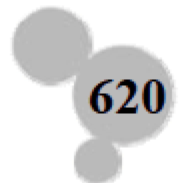
图 27.4 前台首页页面

（1）企业信息

通过“企业管理”页面可对“公司简介”进行修改，进行“规章制度”的添加、修改及删除。

（2）个人计划

通过“个人计划”中的各页面可以添加工作反馈、周计划、月计划、年计划及任务计划。



（3）企业绩效

通过“任务绩效”页面可查看及删除所有员工的年计划和任务计划，通过“质量绩效”页面可查看周计划、月计划及工作反馈信息。

（4）人事消息

通过“消息管理”页面可对公告信息及活动安排信息进行发布、修改及删除。

（5）审核批示

在“发布审核”页面中可以发布、修改及删除审核信息，在“批示审核”页面中可审批及删除审核信息。

（6）考勤管理

通过“上下班登记”、“病事假登记”、“加班登录”页面可添加签到时间，通过“考勤记录”页面可设置考勤时间、查看及删除员工的考勤记录。

（7）职员天地

在“职员浏览”页面中可以查看员工信息；在“个人设定”页面中可修改个人资料；“意见箱”只有部门为“人事部”的员工可以进行回复及删除，其余用户只可以发表意见。

27.3.3　人事消息模块设计

人事消息模块主要是对文件的收发进行管理。模块的设计和实现并不是十分复杂，但在整个系统中的位置却很重要。因为公告栏和意见箱面向的是全体用户，企业的最新动向和职员的意见首先都是在这里体现出来的。

人事消息模块主要包含两部分内容，一部分是面向全体用户的，包括查看公告、活动、意见箱等；一部分是仅对人事部开放，其他用户不允许、也不会看到的页面，如公告管理、意见管理等。人事消息用例图如图 27.5 所示。

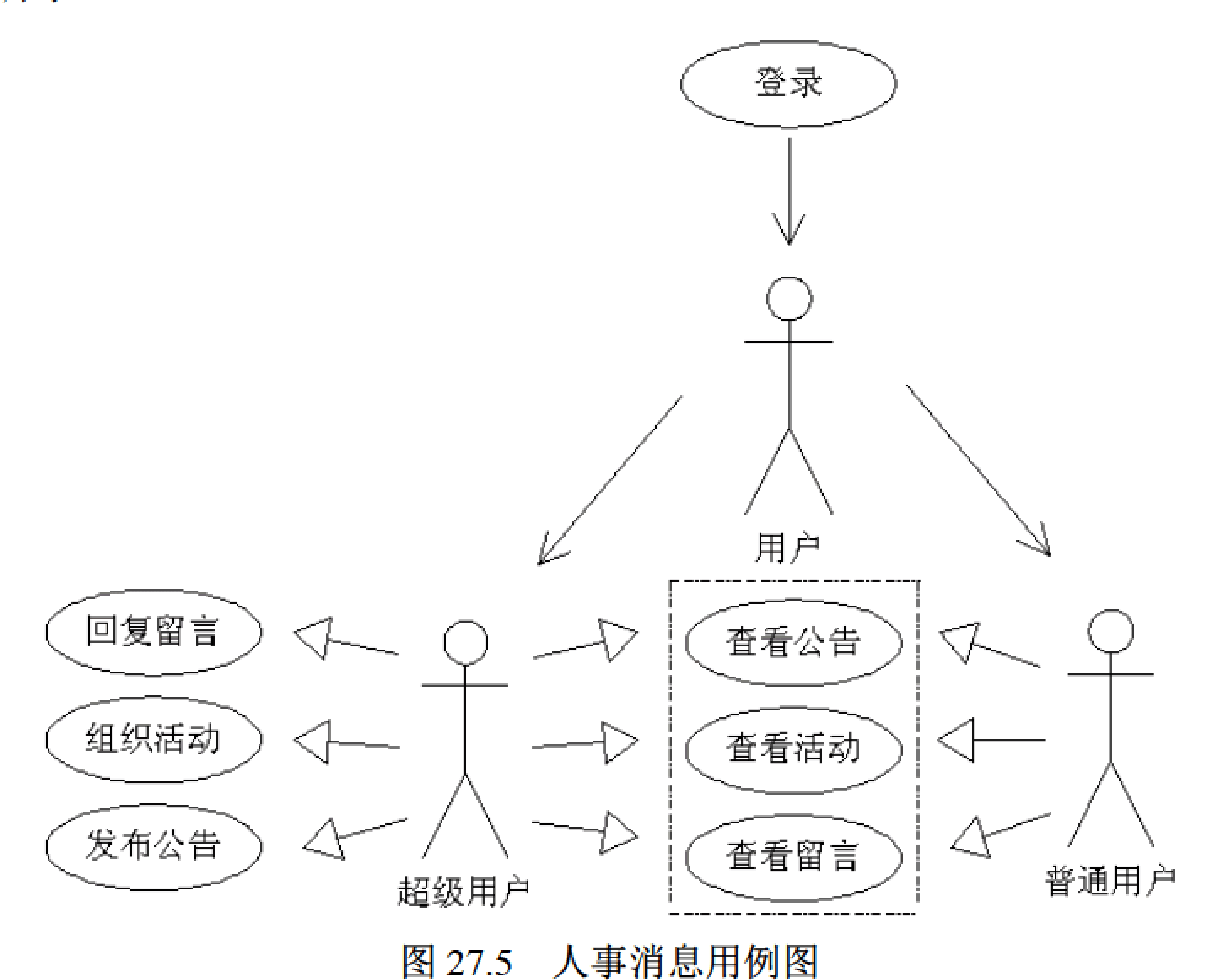


图 27.5　人事消息用例图

27.3.4 考勤管理模块设计

考勤功能是办公自动化管理系统中每天都要使用到的功能模块，也是比较重要的模块之一；除了包括正常的上下班登记功能外，还要包含病事假登记和加班登记等特殊登记功能。管理员通过考勤记录实现对上下班及加班标准时间的设置。

考勤模块主要的功能如下。

☑ 上下班登记：包括上班登记和下班登记。

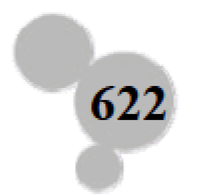
☑ 病事假登记：包括病假登记和事假登记。

☑ 加班登记：包括加班上班登记和加班下班登记。

☑ 考勤记录：包括上下班标准时间设置和加班标准时间设置。

考勤模块的活动图如图 27.6 所示。

图 27.6　考勤模块活动图



27.3.5　后台首页设计

在 IE 地址栏中输入“http://127.0.0.1/tm/sl/27/admin/”，即可访问办公自动化系统的后台登录页面。后台操作页面如图 27.7 所示。后台主要对部门信息、职员信息、用户权限及数据库等信息进行管理。

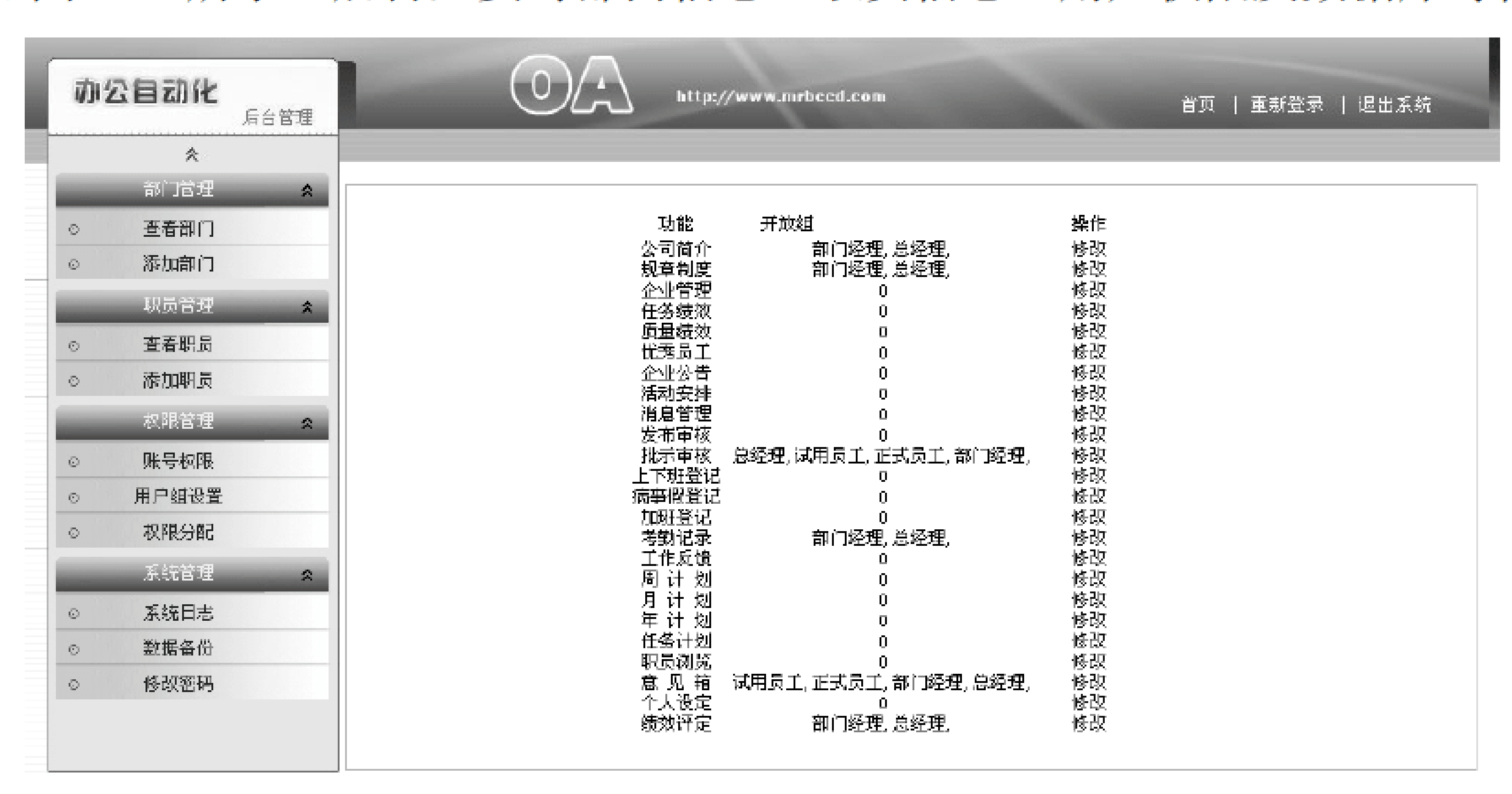


图 27.7　后台操作页面

☑　部门管理：实现部门信息的查看、修改和删除。

☑　职员管理：实现职员信息的查看、修改和删除。

☑　权限管理：设置账号的权限；控制用户属于哪个用户组；控制权限的分配。

注意：当功能没有被指定操作权限组时（就是“开放组”状态为 0 时，如图 27.8 所示），则该功能默认可被所有权限组操作，如图 27.9 所示。但如果该功能被指定操作权限组，则其余权限组将无权操作。而且当新添加权限组时，该功能默认为拒绝操作状态，如图 27.10 所示，如果启用则需要重新添加组。

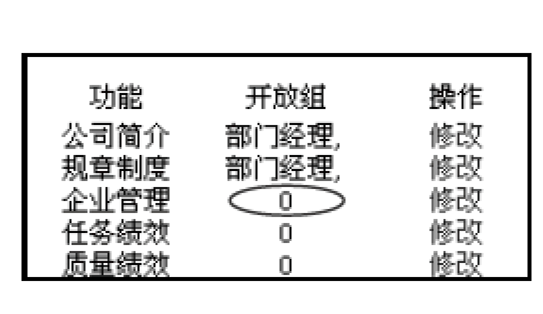


图 27.8　权限分配

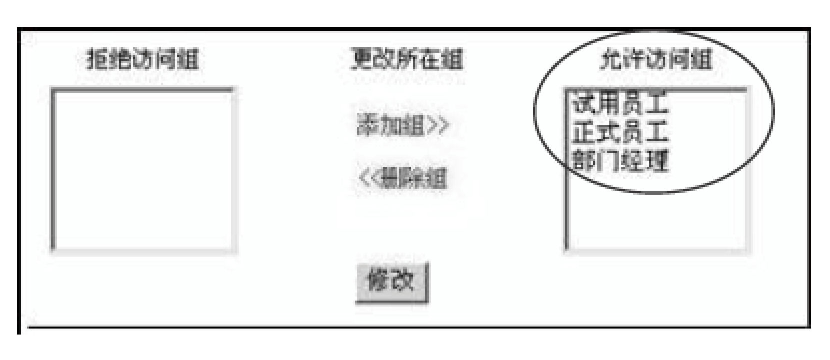


图 27.9　权限分配

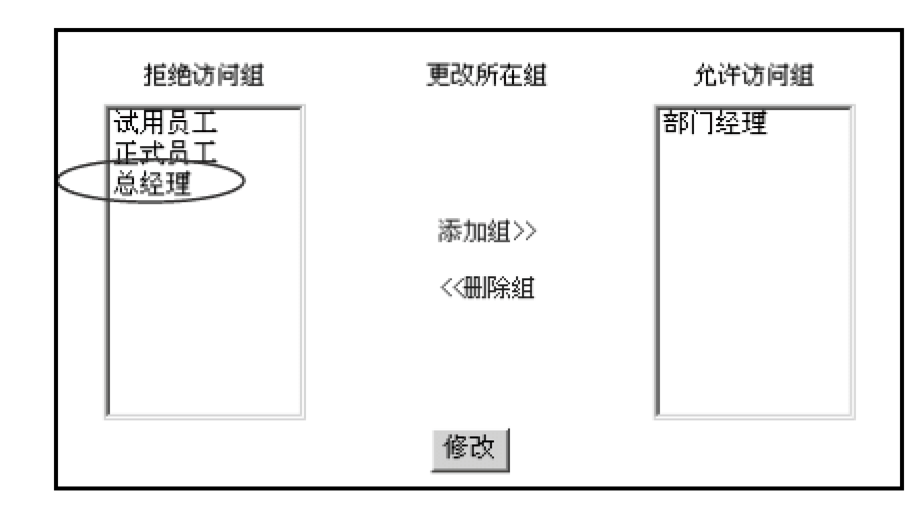


图 27.10　权限分配

☑　系统管理：管理系统日志，备份数据和修改管理员密码。

27.3.6　部门管理模块设计

一个成熟、稳定的企业，一定要有一个合理的部门结构，包括每个部门需要做什么工作、部门之间的协调关系以及部门之间的上下级关系等都要明确下来，这样才能使各部门各尽其职，不会出现资

源浪费，提高企业整体的效率。本系统中部门管理模块包括两大部分：

☑ 查看部门

查看各个部门名称和部门之间的上下级关系，还可以对部门的信息和级别进行修改，如果是最底层的部门，则可以对其进行删除操作。

☑ 添加部门

可以添加新的部门，包括部门名称、上级部门和部门备注。

部门管理操作流程图如图 27.11 所示。

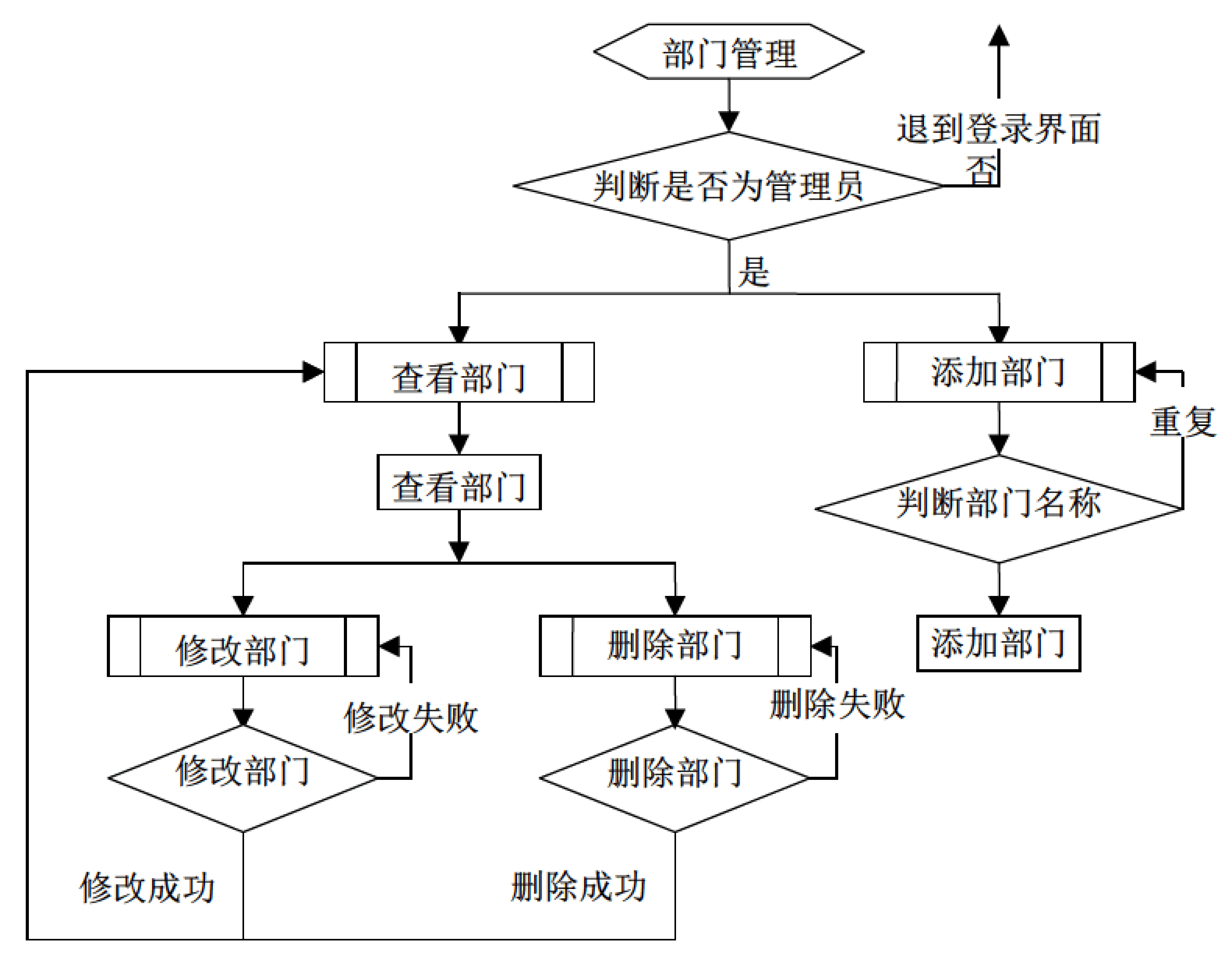
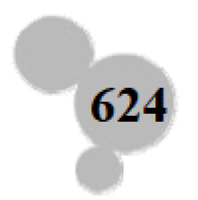


图 27.11　部门管理操作流程图

27.4 本 章 小 结

本章通过一个完整的办公自动化管理系统开发过程，向读者讲述了系统开发的几个重要环节，还包括了几个主要模块的功能实现，最后着重讲解了 MySQL 备份的相关知识。相信通过本章的学习，读者对如何开发一个真实的项目会有进一步的了解，通过自己动手实践，完成一个功能相似的办公自动化管理系统是完全不成问题的。

说明：由于篇幅所限，本章只对办公自动化系统的模块进行了简单介绍，关于办公自动化系统的详细代码可参见本书附带光盘。



第28章

供求信息网

在全球知识经济和信息化高速发展的今天，信息化是决定企业成败的关键因素，企业需要在网站上发布供求信息，才能在同领域的竞争中得到突飞猛进的发展。

一个广泛、快速、自由的信息交流平台，为用户带来方便的同时，也会给企业带来无限的商机。于是，以因特网为基础的信息交流平台的易查供求信息网出现了。易查供求信息网致力于优化信息交流，实现信息的快速交流。

在本系统中应用到的关键技术和方法如下：

- **使当前窗口承载框架页中的超链接页面**
- **自动计算以系统日期为基数的相对日期**
- **do…while 循环语句**
- **查询关键字描红**
- **框架技术在 Web 网站中的应用**
- **表单数据的两种提交方式**
- **通过 MySQl 函数连接、操作数据库**
- **通过 SQL 语句实现数据的查询、更新、修改和删除操作**

- ☑ 服务器端
  - ➢ 操作系统：Windows 2003 Server/Linux（推荐）。
  - ➢ 服务器：Apache 2.2。
  - ➢ PHP 软件：PHP 5.1.6。
  - ➢ 数据库：MySQL 5.0.24。
  - ➢ MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
  - ➢ 开发工具：Dreamweaver 8。
  - ➢ 浏览器：IE 6.0 及以上版本。
  - ➢ 分辨率：最佳效果 1024×768 像素。
- ☑ 客户端
  - ➢ 浏览器：IE 6.0 及以上版本。
  - ➢ 分辨率：最佳效果 1024×768 像素。

28.3　开 发 过 程

28.3.1　数据库设计

本系统是一个中小型的供求信息平台，但是由于平台会涉及海量数据，因此需要充分考虑到成本问题及用于需求（如跨平台）等问题，而 MySQL 是世界上最为流行的开放源码的数据库，是完全网络化的跨平台的关系型数据库系统，这正好满足了中小型企业的需求，所以本系统采用 MySQL 数据库。

结合实际情况及对用户需求的分析，供求信息网中应用的 db_database28 数据库主要包含 4 个数据表，如图 28.3 所示。



服务器: localhost ▸ 数据库: db_database28

表	类型	整理	说明
tb_admin	MyISAM	gb2312_chinese_ci	管理员信息表
tb_advertising	MyISAM	gb2312_chinese_ci	企业广告信息表
tb_info	MyISAM	gb2312_chinese_ci	免费供求信息表
tb_leaguerinfo	MyISAM	gb2312_chinese_ci	付费供求信息表

图 28.3　易查供求信息网数据表

28.3.2　前台功能设计

网站前台关乎着网站的建设及形象宣传，它对网站生存和发展起着非常重要的作用。网站前台应该是一个信息含量较高、内容较丰富的宣传平台。供求信息网前台主要包含以下内容：

- ☑ 网站菜单导航（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等）。
- ☑ 发布免费的供求信息（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等）。
- ☑ 推荐供求信息显示（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等），其中，付费信息按时间顺序降序排列，免费信息按时间顺序分页显示。
- ☑ 显示推荐的企业广告信息。

☑ 供求信息快速检索，支持模糊查询和查询关键字描红功能。

☑ 后台登录入口，为管理员进入后台提供一个入口。

前台功能结构如图 28.4 所示。

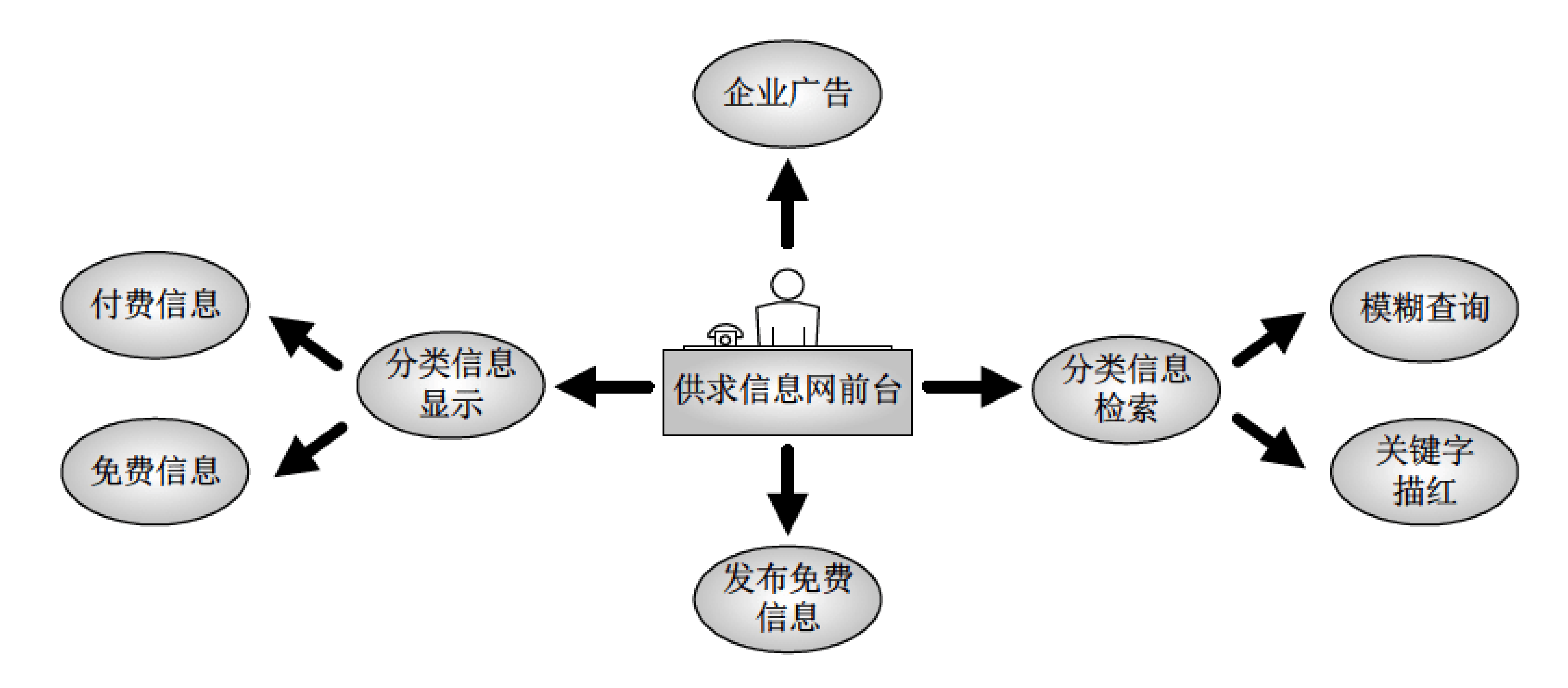


图 28.4　供求信息网前台功能结构图

28.3.3　免费供求信息发布模块设计

免费供求信息发布提供对象为供求信息用户，是供求信息网站非常重要的功能，也是供求信息网站的核心功能。

免费供求信息发布模块可以完成 11 种不同类别信息的发布。用户可以根据自身需要将供求信息发布到相应的信息类别中（公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示）。信息发布成功后，需要管理员进行审核，只有审核成功的信息才能显示在前台相应的信息类别网页中。免费供求信息发布的流程如图 28.5 所示。

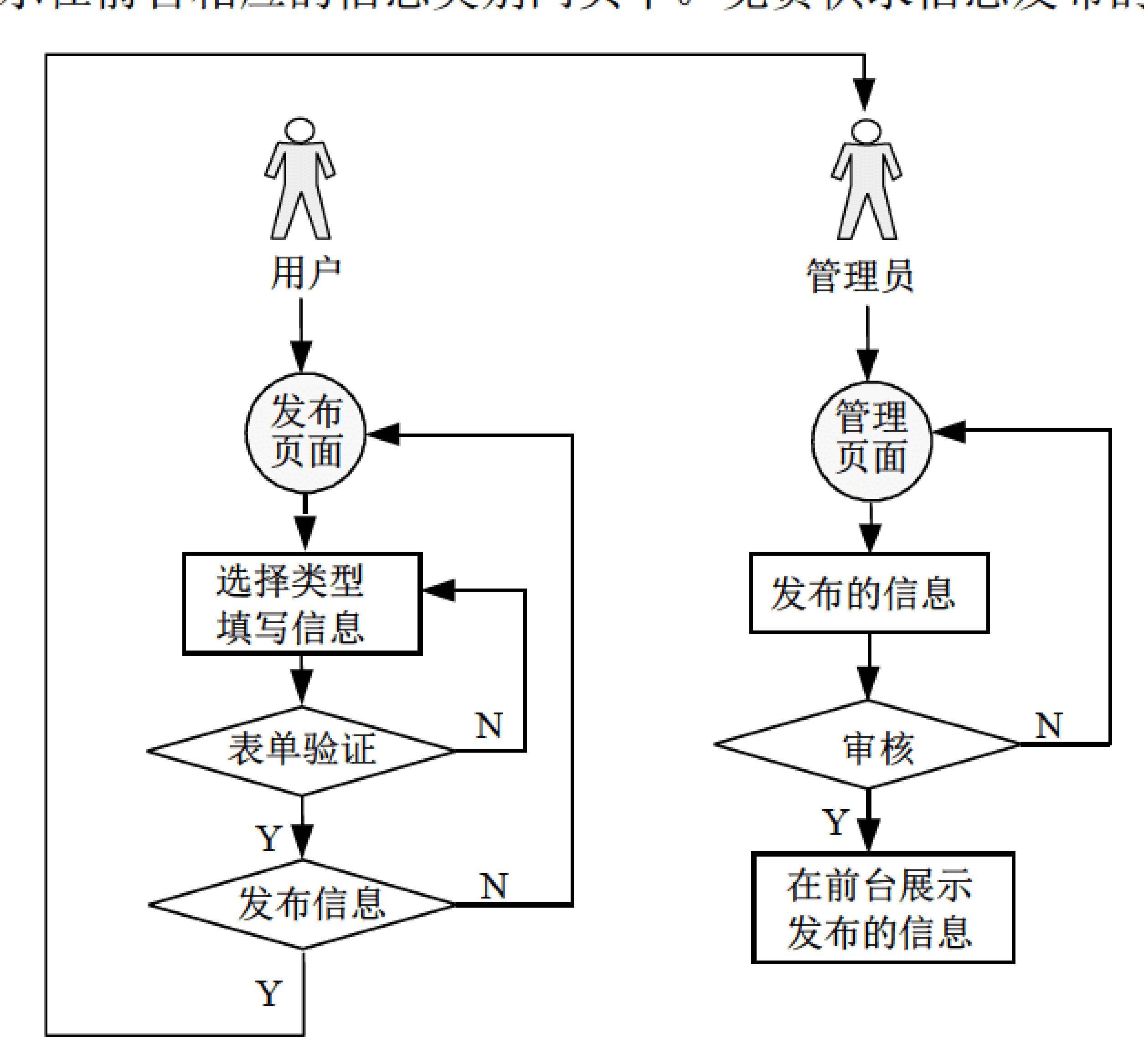
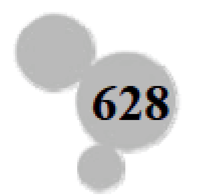


图 28.5　免费供求信息发布流程图



功能创建成功后，用户通过单击前台页面导航栏的“我要发布”超链接，进入信息发布页面，如图 28.6 所示。程序会先来验证用户是否输入信息，若验证失败，则返回信息发布页面，进行相应提示。若验证成功，则向数据库中插入记录，完成发布操作。



图 28.6　免费供求信息发布网页

28.3.4　信息检索模块设计

信息检索是对已存在于数据库中的数据按条件进行筛选浏览，是查看历史信息和确认数据操作最为快速、有效的办法。信息检索模块主要通过选择信息类型和输出查询关键字模糊查询供求信息资源，并输出查询结果。考虑到供求信息的信息量较大，本模块对与查询关键字相匹配的查询结果进行描红，从而方便用户的浏览。信息检索模块示意图如图 28.7 所示。

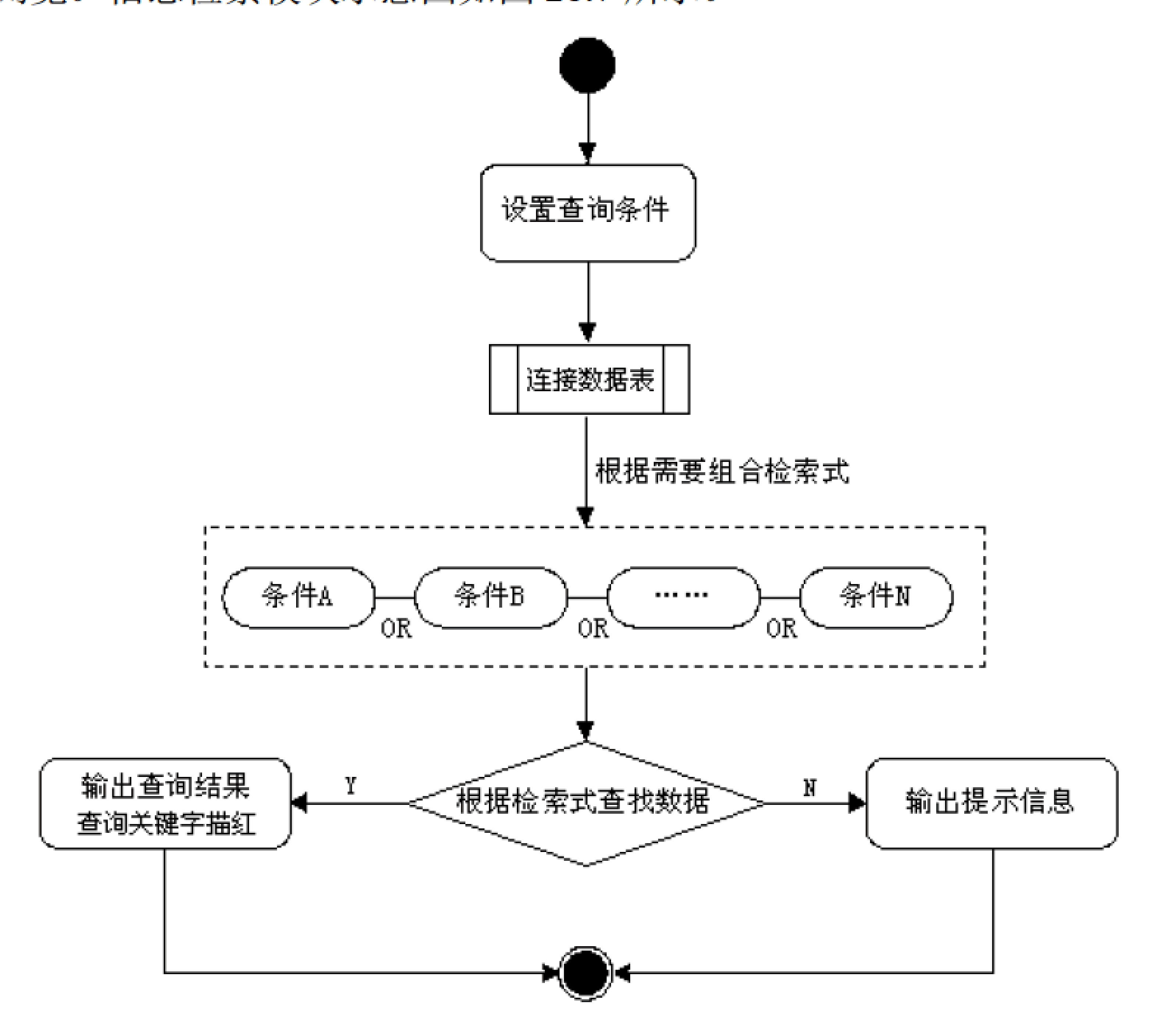


图 28.7　信息检索模块示意图

在开发信息检索模块时，由于该网站含有大量的数据信息，为了方便用户浏览网站信息，需要添加复合条件查询实现搜索功能。在信息检索区的“关键字”文本框中输入欲查询的关键字，在下拉列表框中选择要搜索的信息类型，然后单击“开始搜索”按钮，对指定条件的记录进行检索并输出结果集到浏览器，同时，为了方便浏览者查找自己所关注的内容信息，本模块对查询关键字进行描红。运行结果如图 28.8 所示。



图 28.8　信息检索页面运行结果

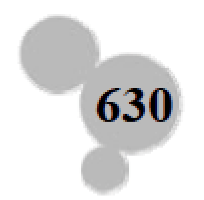
28.3.5　后台功能模块设计

程序开发人员在设计网站后台时，主要考虑功能的易操作性、实用性和易维护性。供求信息网后台主要包含以下内容：

- ☑ 发布付费的供求信息（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等），以及付费信息的浏览、审核及删除功能。
- ☑ 免费信息的浏览、审核及删除功能。
- ☑ 企业广告信息的发布、浏览、前台推荐显示、删除功能。
- ☑ 网站首页，为管理员进入前台提供一个入口。
- ☑ 退出登录，注销当前用户。

其功能结构如图 28.9 所示。

后台管理模块运行结果如图 28.10 所示。



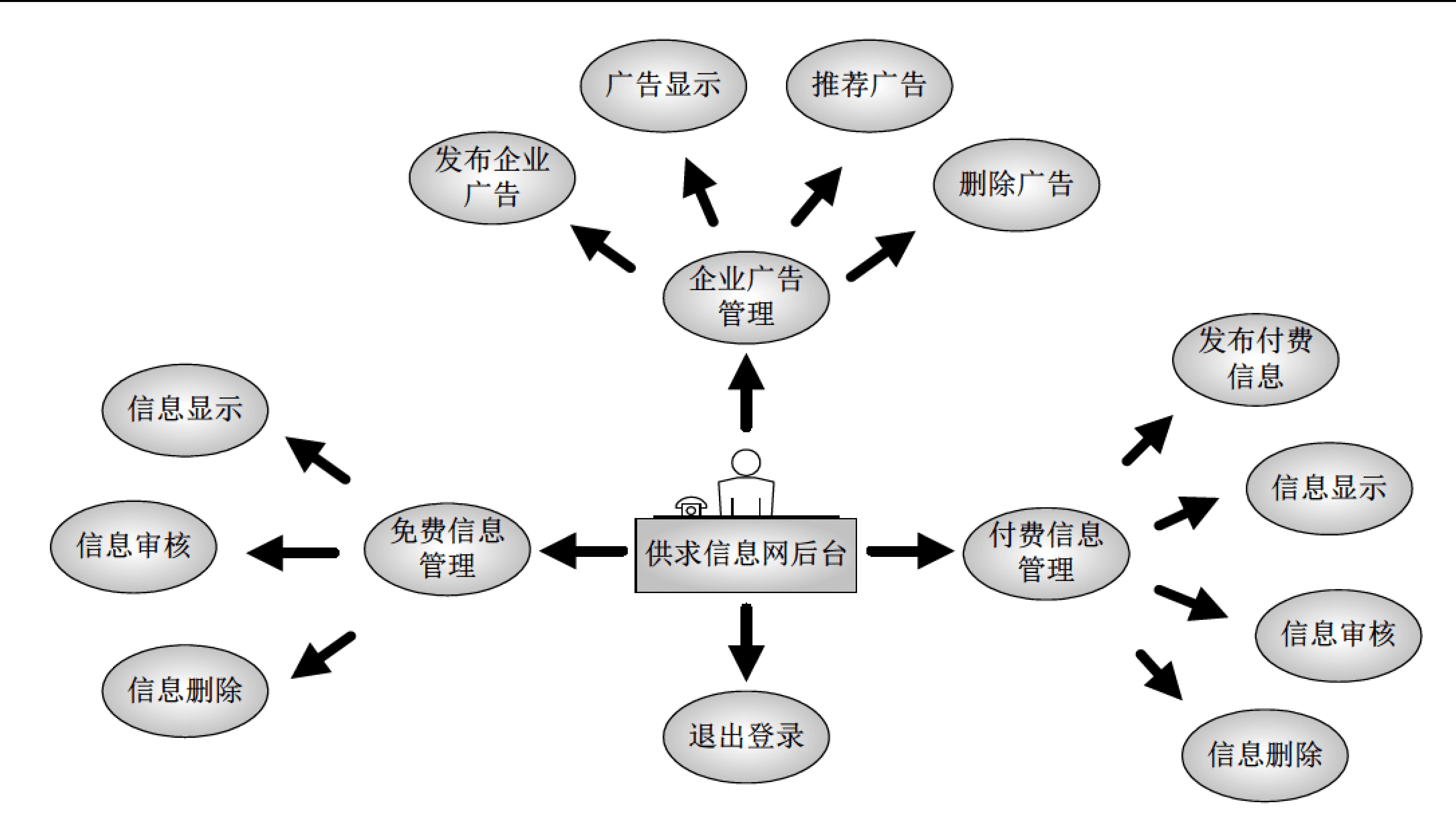


图 28.9　易查供求信息网后台功能结构图



图 28.10　后台管理模块运行结果

28.3.6　付费供求信息发布模块设计

付费供求信息发布的提供对象为供求信息用户，是供求信息网站非常重要的功能，也是供求信息网站的盈利点。企业或用户可以根据自身需要对供求信息先进行付费，付费后由管理员在后台将供求信息发布到相应的信息类别中（招聘信息、求职信息、培训信息、公寓信息、家教信息、车辆信息、物品求购、物品出售、求兑出兑、寻求合作、企业广告）。供求信息成功发布后，管理员需要在后台对发布的供求进行审核，如果审核通过，则显示在前台相应的信息类别网页中。付费供求信息发布的

流程图如图 28.11 所示。

用户通过单击页面导航区的“付费信息”超链接，进入付费信息发布页面，如图 28.12 所示。填写真实有效的付费信息，单击“发布信息”按钮，程序会先来验证用户是否输入的信息，若验证失败，则返回信息发布页面，进行相应提示。若验证成功，则向数据库中插入记录，完成付费信息的发布操作。

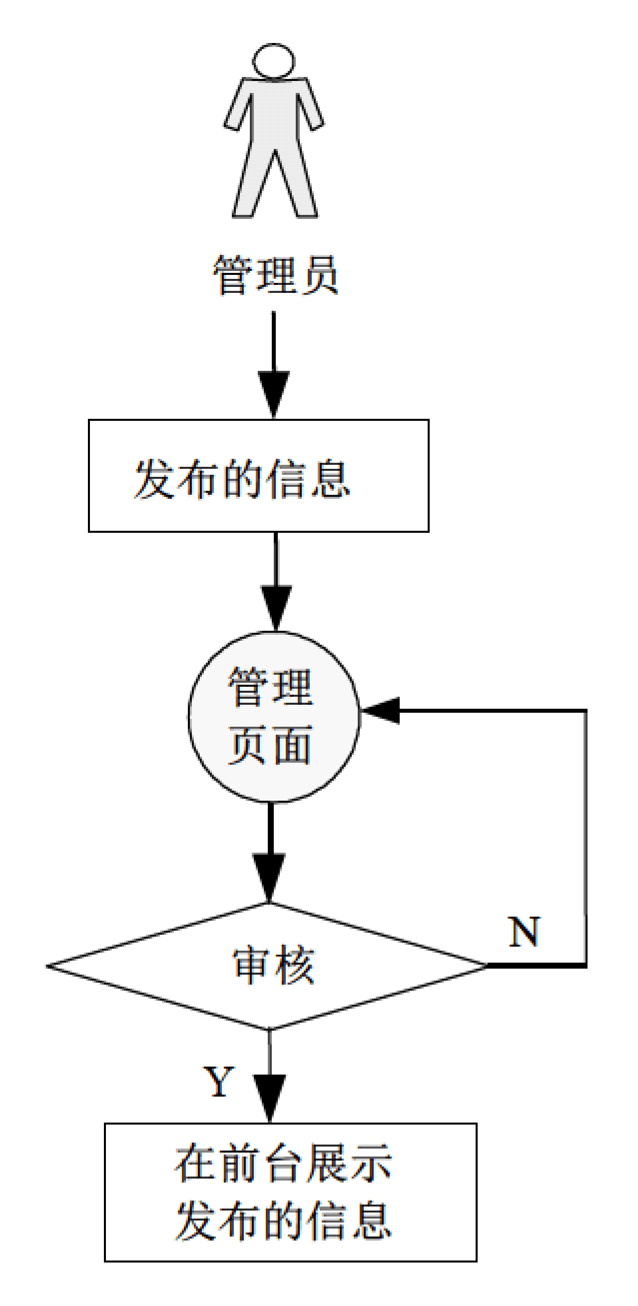


图 28.11　付费供求信息发布流程图



图 28.12　付费供求信息发布页面运行结果

28.4　本章小结

本章依据软件开发流程介绍了易查供求信息网的开发过程。在开发任何一个项目前，首先要充分做好前期准备，如完善的需求分析、清晰的业务流程、合理的程序结构等，这样，在后期的程序开发中才会得心应手、有备而无患。通过本章的学习，读者可以了解供求信息网站的开发流程，熟悉框架技术在 Web 应用程序中的应用。由于篇幅所限，这里没有对该网站的开发进行详细讲解，其具体的功能可参考本书配套光盘中的源代码。

第29章

图书借阅管理系统

在图书馆中，读者可能看到过这样一幅画面：很多人徘徊于图书馆书架之间，找寻着自己需要的图书；图书管理员则翻阅着图书借阅的记录，查看着哪些图书被借出，哪些需要归还，进行着繁琐的统计工作。而当计算机进入到我们的生活以后，可以通过它来管理图书馆中的图书，对图书进行登记、编号、分类；记录图书的借阅信息；查询图书馆中的图书，一切工作都变得那么简单、流畅。不但提高了工作的效率，而且使图书的查询变得方便、快捷，这就是图书借阅管理系统的魅力所在。本章将应用 Apache+PHP+MySQL 开发一个图书借阅管理系统，其中应用到的关键技术和方法如下：

- 通过 MySQL 函数连接数据库的方法
- 通过 PHP 函数操作数据库的方法
- 面向对象的编程方法
- 自动计算图书归还日期
- 管理员权限的设置方法
- MySQL 中添加、修改、删除和更新语句的应用
- MySQL 中多表查询方法
- MySQL 中内联接和外联接语句的应用

29.1 项 目 描 述

知源图书馆是吉林 XX 师范学校的大型图书馆，随着学校图书馆规模的不断壮大，经营图书品种、数量也逐渐增多。同时，此图书馆常年采用的传统的人工方式管理暴露了一些问题。例如，查找读者借阅的某一本图书的具体摆放位置，需要靠人工记忆在书海中苦苦查找，由于图书储存量大，很难准确定位图书的具体位置，因此每天都要浪费大量的人力资源。

学校图书馆为提高工作效率，同时摆脱图书管理人员在工作中出现的种种弊端，现决定开发一个学校图书馆管理系统，通过计算机对图书进行管理。

图书馆管理系统为图书馆注入了新的生机，不但在运营过程中节省了大量的人力、物力、财力和时间，提高了图书馆的效率，而且还为图书馆在读者群中树立了一个全新的形象，为图书馆日后的发展奠定了一个良好的基础。

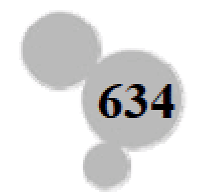
通过对一些大型图书馆的实际考察、分析，并结合图书馆的要求以及实际的市场调查，要求本系统具有以下功能：

- ☑ 系统页面设计美观大方、个性化，功能全面，操作简单。
- ☑ 实现基础信息的管理平台。
- ☑ 统计、管理读者信息。
- ☑ 统计图书借阅排行，了解当前的畅销书。
- ☑ 商品分类详尽，可按不同类别查看图书信息。
- ☑ 提供快速的图书信息、图书借阅检索功能，保证数据查询的灵活性。
- ☑ 实现图书借阅、图书续借、图书归还的功能。
- ☑ 实现综合条件查询，如按用户指定条件查询、按日期时间段查询、综合条件查询等。
- ☑ 实现对图书借阅、续借和归还过程的全程数据信息跟踪。
- ☑ 提供借阅到期提醒功能，使管理者可以及时了解到已经到达归还日期的图书借阅信息。
- ☑ 提供灵活、方便的权限设置功能，使整个系统的管理分工明确。
- ☑ 具备稳定的数据库系统。
- ☑ 安全性高，易于维护。

图书借阅管理系统主页的运行效果如图 29.1 所示。



图 29.1 图书借阅管理系统的主页



图书档案管理页面如图 29.2 所示。

图书借阅操作页面如图 29.3 所示。



图 29.2　图书档案管理页面



图 29.3　图书借阅操作页面

29.2　技 术 准 备

在开发图书借阅管理系统时，该项目所使用的软件及开发环境如下。

☑　服务器端

- 操作系统：Windows 2003 Server/Linux（推荐）。
- 服务器：Apache 2.2。
- PHP 软件：PHP 5.1.6。
- 数据库：MySQL 5.0.24。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- 开发工具：Dreamweaver 8。
- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

☑　客户端

- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

29.3　开 发 过 程

29.3.1　数据库设计

图书借阅管理系统的开发，数据库是必不可少的内容，需要通过数据库来存储图书档案数据、读者档案数据、图书借阅记录数据、图书归还记录数据和管理员数据。从开发的成本和用户需求考虑，

使用 MySQL 数据库是最为合适的选择。

结合实际情况及对用户需求的分析，图书借阅管理系统 db_database29 数据库主要包含 10 个数据表，如图 29.4 所示。

服务器: localhost ▸ 数据库: db_database29

表	类型	整理	说明
tb_bookcase	MyISAM	gb2312_chinese_ci	图书书架信息表
tb_bookinfo	MyISAM	gb2312_chinese_ci	图书信息表
tb_borrow	MyISAM	gb2312_chinese_ci	图书借阅信息表
tb_library	MyISAM	gb2312_chinese_ci	图书馆信息表
tb_manager	MyISAM	gb2312_chinese_ci	管理员信息表
tb_parameter	MyISAM	gb2312_chinese_ci	参数设置信息表
tb_publishing	MyISAM	gb2312_chinese_ci	出版社信息表
tb_purview	MyISAM	gb2312_chinese_ci	权限信息表
tb_reader	MyISAM	gb2312_chinese_ci	读者信息表
tb_readertype	MyISAM	gb2312_chinese_ci	读者类型信息表

图 29.4　图书借阅管理系统数据表

29.3.2　功能模块概述

图书借阅管理系统主要分为 6 个部分，每个部分中包括多个具体的功能模块。图书借阅管理系统的整体功能结构如图 29.5 所示。

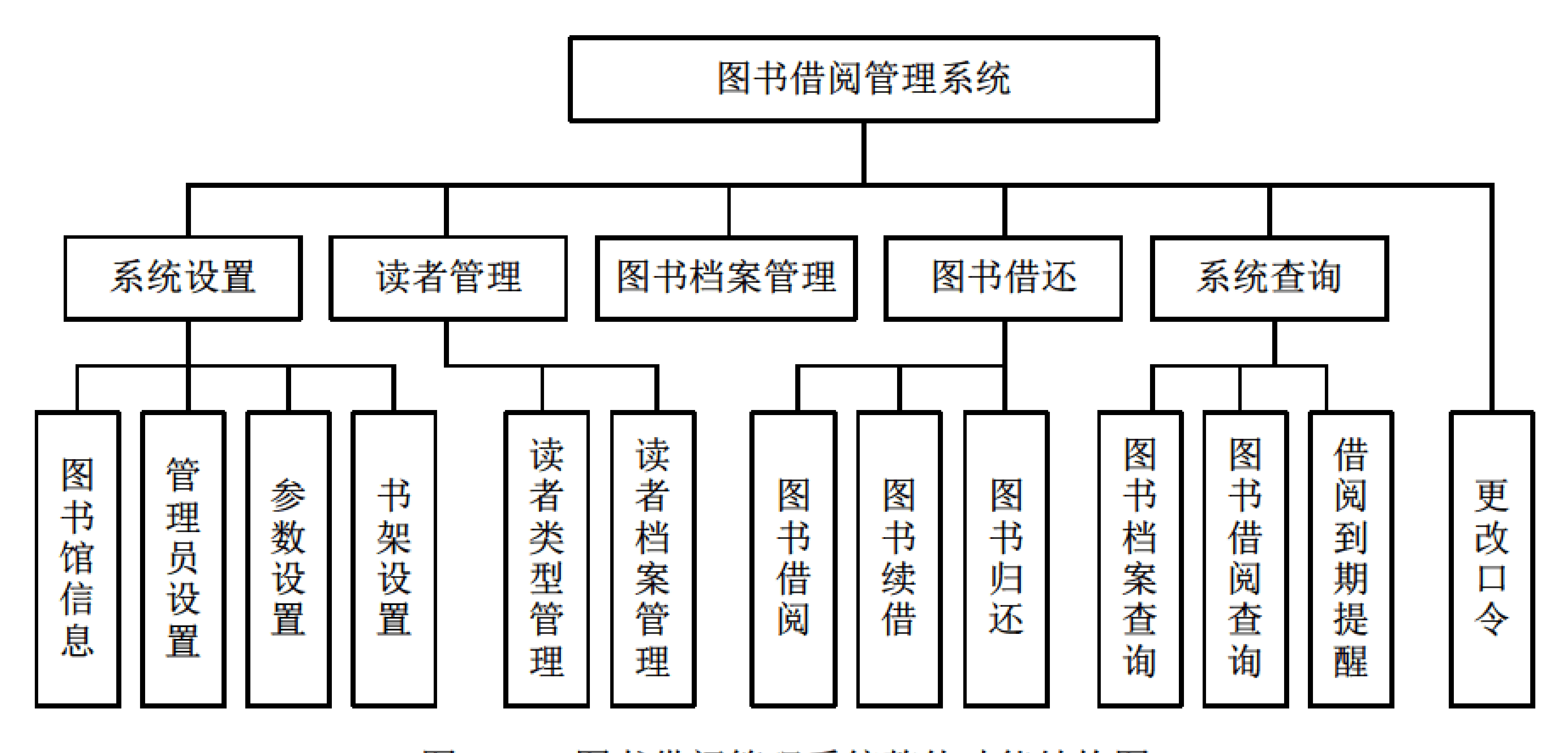


图 29.5　图书借阅管理系统整体功能结构图

29.3.3　管理员模块

管理员模块的功能包括管理员登录、查看管理员列表、添加管理员信息、管理员权限设置、管理员删除和更改口令等。管理员模块的框架图如图 29.6 所示。

首先是管理员登录，用于验证管理员的身份。运行本系统，首先进入的是管理员登录页面，在该页面中，系统管理员可以通过输入正确的管理员名称和密码登录到系统首页，当用户没有输入管理员名称或密码时，系统会通过 JavaScript 进行判断，并给予提示信息。管理员登录页面的运行结果如图 29.7 所示。



管理员登录后，选择“系统设置”/“管理员设置”命令，进入到查看管理员列表页面，页面中将以表格的形式显示全部管理员及其权限信息，并提供添加管理员信息、删除管理员信息和设置管理员权限的超链接。管理员列表页面的运行结果如图 29.8 所示。

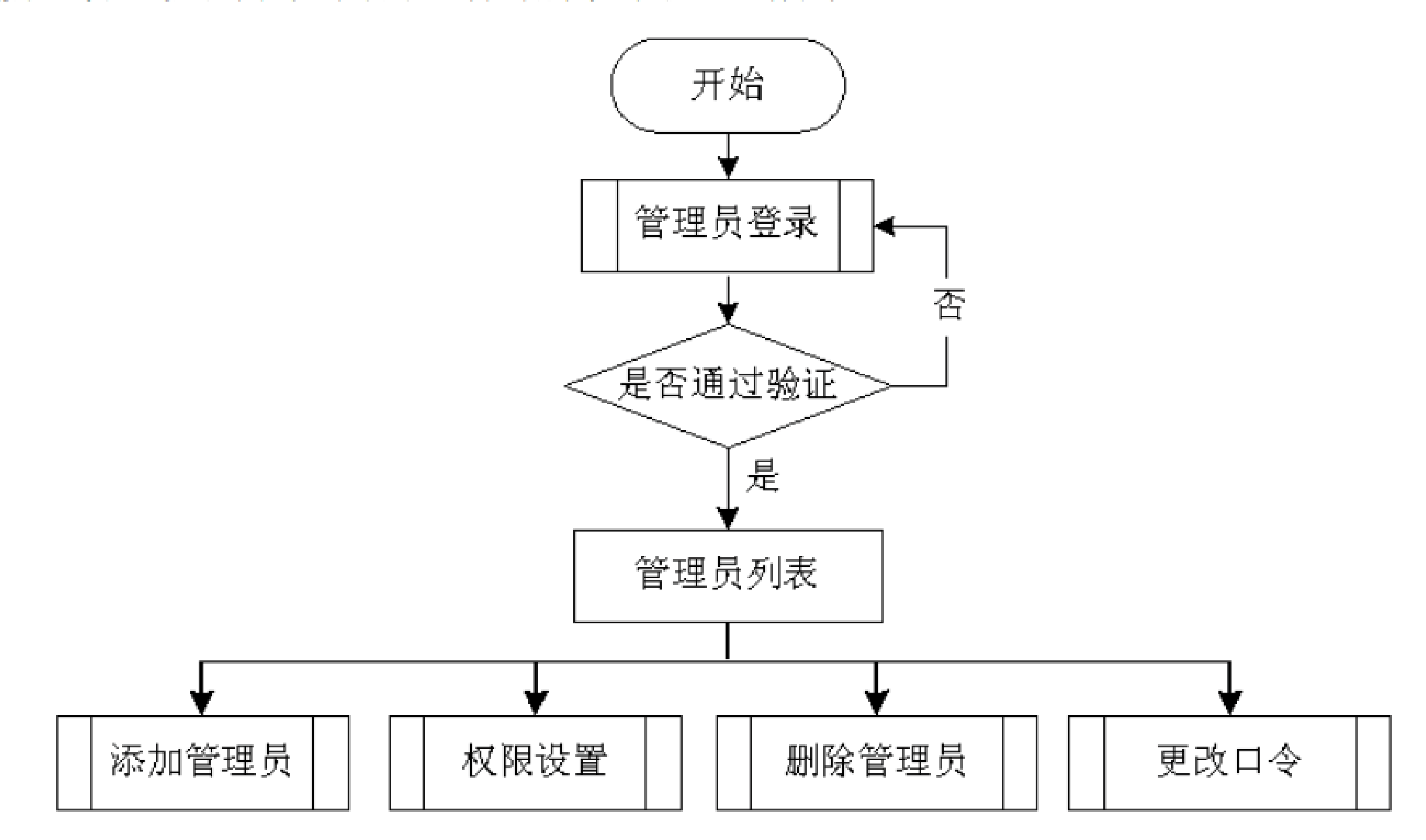


图 29.6　管理员模块的框架图



图 29.7　管理员登录页面的运行结果



图 29.8　管理员列表页面的运行结果

29.3.4 图书档案管理模块

图书档案管理模块的功能包括查看图书列表、添加图书信息、修改图书信息、删除图书信息和查看图书详细信息等。图书档案模块的框架图如图 29.9 所示。

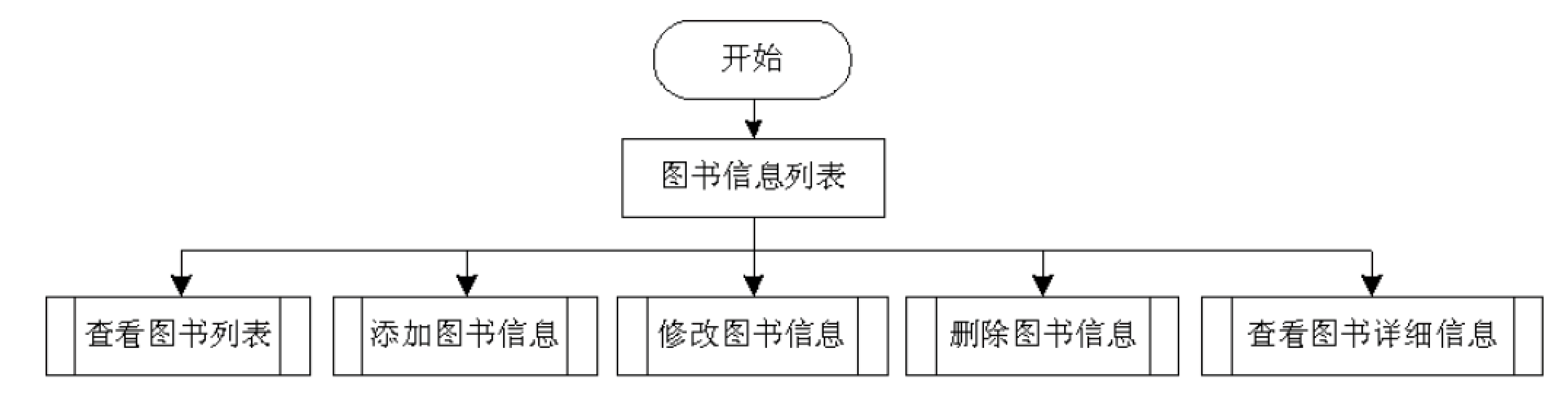


图 29.9 图书档案模块的框架图

管理员登录后，选择“图书管理”/“图书档案管理”命令，进入到查看图书列表页面，在该页面中将显示全部图书信息列表，同时提供添加图书信息、删除图书信息、修改图书信息的超链接。图书信息列表页面的运行结果如图 29.10 所示。

知源 图书馆管理系统

当前登录的用户：Tsoft

2009年5月18日 9:14:36 首页 | 系统设置 | 读者管理 | 图书档案管理 | 图书借还 | 系统查询 | 更改口令 | 注销

当前位置：图书管理 > 图书档案管理 >>>

添加图书信息

条形码	图书名称	图书类型	出版社	书架	修改	删除
123456789	PHP数据库系统开发完全手册	数据库技术	人民邮电出版社	PHP书架	修改	删除
123454321	PHP程序开发范例宝典	宝典系列	人民邮电出版社	PHP书架	修改	删除
987654321	PHP函数参考大全	信息科学技术	人民邮电出版社	PHP书架	修改	删除
9787115154101	Visual Basic控件参考大全	计算机程序设计	人民邮电出版社	VB书架	修改	删除

CopyRight © 2007 www.mrbccd.com 吉林***师范大学图书馆

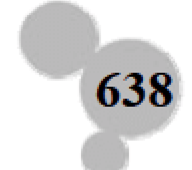
本站请使用IE 6.0或以上版本 1024*768为最佳显示效果

图 29.10 图书信息列表的运行结果

29.3.5 图书借还模块

图书借还模块主要包括图书借阅、图书续借、图书归还、图书档案查询、图书借阅查询、借阅到期提醒等功能。在图书借阅模块中的用户只有一种身份，即操作员，通过该身份可以进行图书借还等相关操作。图书借还模块的用例图如图 29.11 所示。

管理员登录后，选择“图书借还”/“图书借阅”命令，进入到图书借阅页面，在该页面中的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的图书借阅信息。如果检索到对应的读者信息，将其显示在页面中，此时输入图书的条形码或图书名称后，单击“确定”按钮，即可借阅指定的图书。图书借阅页面的运行结果如图 29.12 所示。



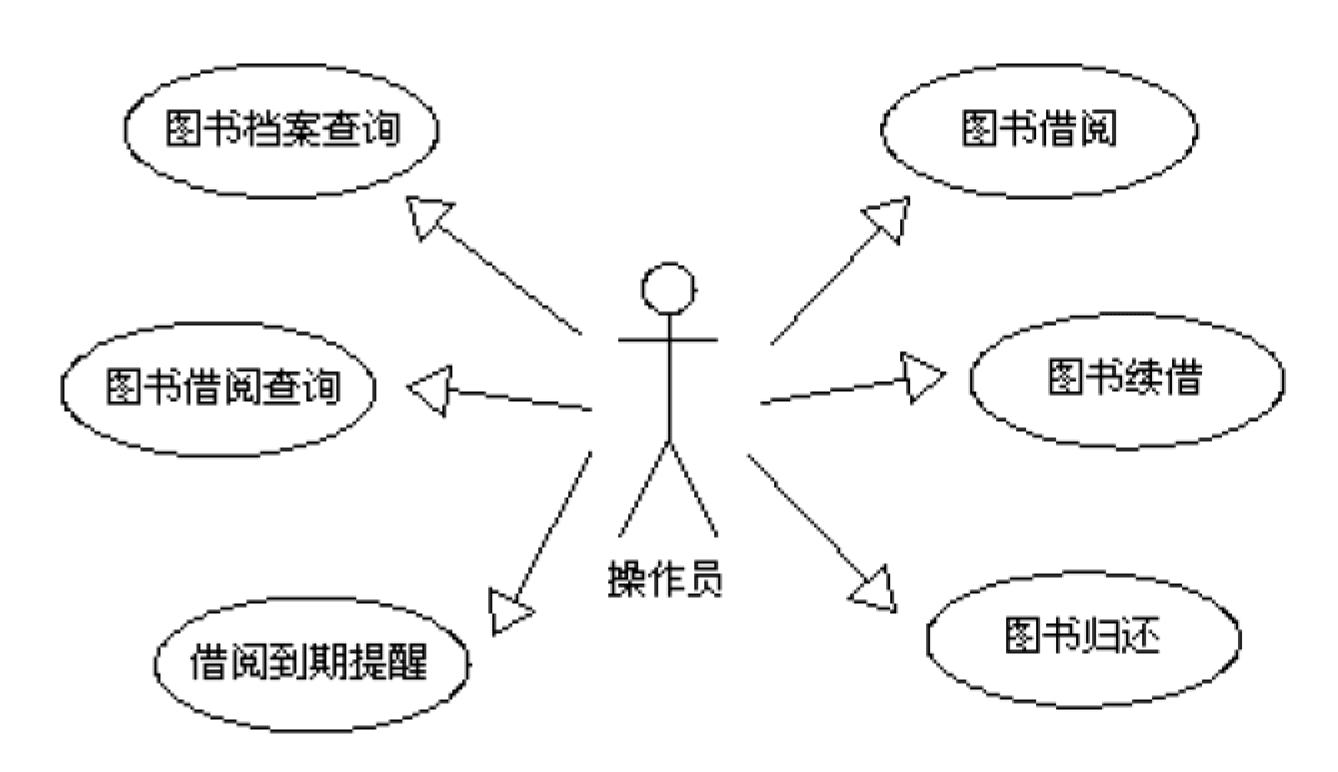


图 29.11　图书借还模块的用例图



图 29.12　图书借阅页面的运行结果

管理员登录后，选择“图书借还”/“图书续借”命令，进入到图书续借页面，在该页面中的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的图书借阅信息。如果检索到对应的读者信息，则将其显示在页面中，此时单击“续借”超链接，即可续借指定图书（将该图书的归还时间加上该书的可借天数得出）。图书续借页面的运行结果如图 29.13 所示。



图 29.13　图书续借页面的运行结果

附录 A　PHP 编码规则

PHP 编码规则不是一个定理、一个公式，不需要 PHP 程序员去刻意遵循，但是如果 PHP 程序员遵循 PHP 编码规则，就可以提高代码的质量、程序的可维护性，及开发速度和效率。如果是团队开发项目，还可以让参与项目的其他开发人员更容易了解项目中的代码，弄清程序的状况，更快地投入到项目开发的工作中来。

如果是闭门造车，形成一套自己的编码风格，就会导致编写的东西只有自己才能看懂，其他人在阅读时会浪费过多的时间和精力，不易于对程序的维护、更新。如果是团队开发项目，就很难投入到项目的开发工作中，因为其他人不了解这个编码规则，也就不能快速地阅读相应的程序代码。

A.1　PHP File 文件格式

A.1.1　常规

对于只包含 PHP 代码的文件，结束标志（"?>"）是不允许存在的，否则会导致文件末尾被意外地注入空白并显示输出。

由__HALT_COMPILER()允许的任意的二进制代码的内容被 Zend Framework PHP 文件或由它们产生的文件禁止。这个功能只对特殊的安装脚本开放。

A.1.2　缩进

使用 4 个空格的缩进，而不使用制表符 TAB。

A.1.3　行的最大长度

一行在 80 个字符以内比较合适，长些也可以，但最多为 120 个字符。

A.1.4　行结束标志

行结束标志只能是标准的 UNIX 文本文件的换行，换行符在文件中表示为 10，或十六进制的 0x0A。

不要使用 Macintosh 的回车，如（0x0D）。

不要使用 Windows 的回车换行组合，如（0x0D,0x0A）。

程序块划分尽量合理，过大或者过小的分割都会影响他人对代码的阅读和理解，一般以较大函数定义、逻辑结构、功能结构来进行划分。少于 15 行的程序块，可不加上下空白行。

说明或显示部分中，内容如中文、数字、英文单词混杂，应当在数字或者英文单词的前后加入空格。

符合上述编码规范的示例代码如下：

```
$result = (($a + 1) * 3 / 2 + $num)).'Test';
$condition ? func1($var) : func2($var);
$condition ? $long_statement
: $another_long_statement;
if ($flag)
{

//Statements
//More than 15 lines
}
Showmessage(‘请使用 restore.php 工具恢复数据。’);
```

A.2　命名约定

A.2.1　类

类名只允许有字母和数字字符，但不鼓励使用数字。下划线只允许作为路径分隔符，如 Zend/Db/Table.php 文件中对应的类名称是 Zend_Db_Table。

如果类名包含多个单词，每个单词的第一个字母必须大写，连续的大写是不允许的，例如，Zend_PDF 是不允许的，而 Zend_Pdf 是可接受的。

由 Zend 或其参与 Zend Framework 项目的伙伴公司发行的类必须以 Zend_开头并且必须按等级顺序放在 Zend/目录下。

A.2.2　接口

接口类也必须遵循同样的约定（如上所述），但必须以 Interface 结尾，例如：

```
Zend_Log_Adapter_Interface
Zend_Controller_Dispatcher_Interface
```

A.2.3　文件名

对于其他文件，只有字母和数字字符、下划线和短横线“-”可用，空格是不允许的。

包含任何 PHP 代码的任何文件必须以.php 扩展名结尾。

普通程序和能够被 URL 直接调用的程序，如 list.php、index.php，可以直接使用“程序名+.php”

的方式命名。

函数库和类库程序，应以小写.func.php 和.class.php 作为扩展名。函数库和类库程序只能被其他程序引用，而不能独立运行。其中，不能包含任何流程性的、不属于任何函数或类的程序代码。

流程性程序，以小写.inc.php 作为扩展名。只能被其他程序引用，而不能独立运行。其中，不能包含任何函数或类代码的程序代码。

模板源文件，以小写.html 作为扩展名。模板源文件按照 PHP 模板编码规则进行编写，不是可执行的程序，而只能被 PHP 模板编译器所解析，放置于./templates/default 或./templates 目录下。

模板目标文件，模板文件被编译后自动生成的目标程序，以小写.php 作为扩展名，存放于./data/templates 目录下。

语言包文件，以小写.lang.php 作为扩展名，只能存放模板或程序使用的语言包信息。

缓存文件，此类文件为系统自动生成，以 cache_xxx.php、usergroup_xxx.php、style_xxx.php 等类似形式命名，存放于./data/cache 目录下。

文件名必须遵循上述的对应类名的规则。

A.2.4　函数和方法

函数名只能包含字母和数字字符，但不鼓励使用数字，不允许使用下划线。

函数名总是以小写字母开头，当函数名包含多个单词时，每个单词的首字母必须大写，这就是所谓的“驼峰”格式。

鼓励使用冗长的名字，这样容易理解代码。下面是符合约定的命名：

```
filterInput()
getElementById()
widgetFactory()
```

对于面向对象编程，对象的访问器总是以 get 或 set 为前缀。当使用设计模式如单态模式（singleton）或工厂模式（factory）时，方法的名字应当包含模式的名字，这样容易从名字识别设计模式。

在对象中的方法，声明为 private 或 protected 的，名称的首字符必须是一个单独的下划线，这是唯一的下划线在方法名字中的用法。声明为 public 的方法不以下划线开头。

允许但不鼓励全局函数（floating functions），建议把这类函数封装到静态类中。

以标准计算机英文为蓝本，尽量避免使用拼音或拼音英文混合的命名方式。

函数的名字和变量的命名规范一致。

函数定义中的左小括号与函数名之间无空格。

开始的左大括号另起一行。

具有默认值的参数应该位于参数列表的后面。

函数定义与调用的参数之间加入一个空格。

尽量避免函数起始缩进位置与结束缩进位置不同。

符合上述书写规则的函数定义方法如下：

```
function authcode($string, $operation, $key = '')
{
```



```
if($flag)
{
//Statement
}
//函数体
}
```

A.2.5　变量

变量只包含数字和字母字符，不鼓励使用数字，不允许为下划线。

声明为 private 或 protected 的类成员变量名必须以一个单独的下划线开头，这是唯一的下划线在变量名中的用法。声明为 public 的变量从不以下划线开头。

与函数名一样，变量名总以小写字母开头并遵循“驼峰式”命名约定。

鼓励使用冗长的名字，这样容易理解代码。除在小循环中以外，不鼓励使用简洁的名字（如$i 和$n）。如果一个循环超过 20 行代码，索引的变量名必须是具有描述意义的名字。

变量命名只能使用项目中有据可查的英文缩写方式，例如，可以使用$data 而不可使用$data1、$data2 这样容易产生混淆的形式，应当使用$username、$password 这样能使人一目了然容易理解的形式。

可以合理地对过长的命名进行缩写，如$bio($biography)、$tpp($threadsPerPage)，前提是英文中存在这样既有的缩写形式，或字母符合英文缩写规范。

必须清楚所使用英文单词的词性，在权限相关的范围内，大多使用$enable***、$is***的形式，前者后面接动词，后者后面接形容词。

下面是一些基本的命名规则：

☑　名称的开头不能使用数字及特殊符号。

☑　“・”及类型声明等专用语不能作为名称。

☑　变量或程序名的长度必须在 255 个字符以内。

☑　和保留字相同的名称不能使用。

A.2.6　常量

常量包含数字、字母和下划线，数字允许作为常量名。

常量名的所有字母必须大写。

为加强可读性，常量中的单词必须以下划线分隔，例如，可以使用 EMBED_SUPPRESS_EMBED_EXCEPTION，但是不允许使用 EMBED_SUPPRESSEMBEDEXCEPTION。

常量必须通过 const 定义为类的成员，不鼓励使用 define 定义的全局常量。

A.2.7　数据库表和字段

表和字段的命名以命名原则中定义的规范为依据。

所有数据表名称，只要其名称是可数名词，应尽量以复数方式命名；存储多项内容的字段，或代

表数量的字段，也应当以复数方式命名，如 hits（查看次数）、items（内容数量）。

当几个表间的字段有关联时，要注意表与表之间关联字段命名的统一，如 forum_articles 表中的 articleid 与 forum_restores 表中的 articleid。

代表 ID 自增量的字段，通常用以下几种形式：

☑ 一般情况下，使用全称的形式，如 userid、articleid。

☑ 没有功能性作用，只为管理和维护方便而设的 ID，可以使用全称的形式，也可只将其命名为 ID。

A.3 编 码 风 格

A.3.1 PHP 代码划分（Demarcation）

PHP 代码总是用完整的标准的 PHP 标签定界：

```
<?php

?>
```

短标签（<? ?>）是不允许的，只包含 PHP 代码的文件不要结束标签。

A.3.2 字符串

1．字符串文字

如果字符串是文字（不包含变量），则用单引号（apostrophe）引起来：

```
$a = 'Example String';
```

2．包含单引号（'）的字符串文字

如果文字字符串包含单引号（apostrophe），就用双引号引起来，特别是在 SQL 语句中：

```
$sql = "SELECT 'id', 'name' from 'people' WHERE 'name'='Fred' OR 'name'='Susan'";
```

在转义单引号时，上述语法是首选。

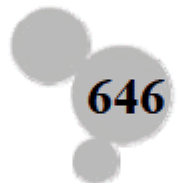
3．变量替换

变量替换有下面两种形式：

```
$greeting = "Hello $name, welcome back!";
$greeting = "Hello {$name}, welcome back!";
```

为保持一致，下面的形式是不允许的：

```
$greeting = "Hello ${name}, welcome back!";
```



码。示例如下：

```
/**
 * Documentation Block Here
 */
class ExampleClass
{
    // entire content of class
    // must be indented four spaces
}
```

2．类成员变量

必须用下面的变量名约定来命名类成员变量：

☑ 变量的声明必须在类的顶部，要先于方法的声明。

☑ 不允许使用 var，应使用 private、protected 或 public。

A.3.5 函数和方法

1．函数和方法声明

必须用下面的变量名约定来命名函数：

☑ 在类中的函数必须用 private、protected 或 public 声明其可见性。

☑ 和类一样，花括号从函数名的下一行开始，函数名与包括参数的圆括号中间没有空格。

☑ 不能使用全局函数。

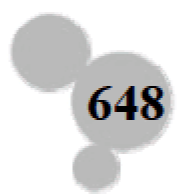
示例如下：

```
/**
 * Documentation Block Here
 */
class Euspice
{
    /**
     * Documentation Block Here
     */
    public function cattle()
    {
        // entire content of function
        // must be indented four spaces
    }
}
```

2．函数和方法的用法

函数的参数用逗号和空格将紧接着的参数分隔。下面例子中的函数带有 3 个参数：

```
threeArguments(1, 2, 3);
```



传值方式在调用时是禁止的。

带有数组参数的函数，函数的调用可包括 array 并分成多行来提高可读性，同时，书写数组的标准仍然适用：

```
threeArguments(array(1, 2, 3), 2, 3);
threeArguments(array(1, 2, 3, 'Zend', 'Studio',
                     $a, $b, $c,
                     56.44, $d, 500), 2, 3);
```

A.3.6　控制语句

1．if/else/elseif

使用 if/else/elseif 的控制语句时，在条件语句的圆括号前后都必须有一个空格。

在圆括号中的条件语句，操作符必须用空格分开，鼓励使用多重圆括号以便在复杂的条件中划分逻辑组合。

前花括号必须和条件语句在同一行，后花括号单独在最后一行，其中的内容用 4 个空格缩进。

```
if ($a != 2) {
    $a = 2;
}
```

下面的例子应用 if、else 和 elseif，符合上述格式约定：

```
if ($a != 2) {
    $a = 2;
} else {
   $a = 7;
}
if ($a != 2) {
    $a = 2;
} elseif ($a == 3) {
   $a = 4;
} else {
   $a = 7;
}
```

在有些情况下，PHP 允许这些语句不用花括号，但在 ZF 代码标准里，if、elseif 或 else 语句必须使用花括号。

elseif 是允许的但强烈不鼓励使用，最好使用 else if 组合。

2．switch

在 switch 结构的控制语句中，条件语句的圆括号前后都必须有一个单独的空格。

switch 中的代码必须有 4 个空格缩进，在 case 中的代码再缩进 4 个空格。

```
switch ($numPeople) {
    case 1:
```

```
            break;

        case 2:
            break;

        default:
            break;
}
```

switch 语句中必须有 default。

A.3.7 数据库

1. 字段结构

允许 NULL 值的字段，数据库在进行比较操作时，会先判断其是否为 NULL，非 NULL 时才进行值的比较。因此基于效率的考虑，所有字段均不能为空。

2. SQL 语句

所有 SQL 语句中，除了表名、字段名称以外，全部语句和函数均需大写，应当杜绝小写形式或大小写混杂的写法，如“select * from tb_user;”是不符合规范的写法。

长 SQL 语句应当有适当的断行，依据 JOIN、FROM、ORDER BY 等关键字进行界定。

通常情况下，在对多表进行操作时，要根据不同表的名称，对每个表指定一个 1～2 个字母的缩写，以使语句简洁并提高可读性。下面的 SQL 语句是符合编码规范的：

```
$result = $db->query("Select m.*, i.*
FROM ".TABLE_MEMBER." m, ".TABLE_MEMBERINFO." i
Where m.userid=i.userid AND m.userid='$_userid'");
```

3. 定长表与变长表

包含任何 varchar、text 等变长字段的数据表，即为变长表，反之则为定长表。

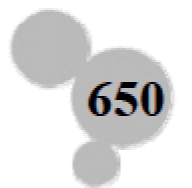
对于变长表，由于记录大小不同，在其上进行许多删除和更改将会使表中的碎片更多，需要定期运行 OPTIMIZE TABLE 以保持性能，而定长表就没有这个问题。

如果表中有可变长的字段，将它们转换为定长字段能够改进性能，因为定长记录易于处理。但在这样做之前，应该考虑下列问题：

- ☑ 使用定长列速度更快，但占用的空间更多。char(n)类型列的每个值总要占用 n 个字节（即使空串也是如此），因为在表中存储时，值的长度不够将在右边补空格。
- ☑ varchar(n)类型的列所占空间较少，因为只给它们分配存储每个值所需要的空间，每个值再加一个字节用于记录其长度。因此，如果在 char 和 varchar 类型之间进行选择，需要对时间与空间作出折衷。

变长表到定长表的转换，不能只转换一个可变长字段，必须对它们全部进行转换，而且必须使用一个 Alter TABLE 语句同时全部转换，否则转换将不起作用。

有时不能使用定长类型，如对于比 255 字符更长的字串就没有定长类型。



在设计表结构时如果能够使用定长数据类型尽量用定长的，因为定长表的查询、检索、更新速度都很快。必要时可以把部分关键的、承担频繁访问的表进行拆分。

进行表结构设计时，应当做到恰到好处、反复推敲，从而实现最优的数据存储体系。

4. 结构优化与索引优化

索引能加快查询速度，而索引优化和查询优化是相辅相成的，既可以依据查询对索引进行优化，也可以依据现有索引对查询进行优化，这取决于修改查询或索引哪个对现有架构和效率的影响最小。

索引优化与查询优化的最基本准则如下：

☑ 根据产品的实际运行和被访问情况，找出哪些 SQL 语句是最常被执行的。最常被执行和最常出现在程序中是完全不同的概念。最常被执行的 SQL 语句，又可被划分为对大表（数据条目多的）和对小表（数据条目少的）的操作。无论大表或小表，又可分为读（Select）多、写（Update/Insert）多或者读写都多的操作。

对常被执行的 SQL 语句而言，对大表操作需要注意：

- 写操作多的，通常可使用写入缓存的方法，先将需要写或需要更新的数据缓存至文件或其他表，定期对大表进行批量写操作。同时，应尽量使得常被读写的大表为定长类型，即使原本的结构中大表并非定长。大表定长化，可以通过改变数据存储结构和数据读取方式，将一个大表拆成一个读写多的定长表和一个读多写少的变长表来实现。
- 读操作多的，需要依据 SQL 查询频率设置专门针对高频 SQL 语句的索引和联合索引。

而小表就相对简单，加入符合查询要求的特定索引，通常效果比较明显。同时，定长化小表也有益于效率和负载能力的提高。字段比较少的小定长表，甚至可以不需要索引。

☑ 看 SQL 语句的条件和排序字段是否动态性很高（即根据不同功能开关或属性，SQL 查询条件和排序字段的变化很大的情况），动态性过高的 SQL 语句是无法通过索引进行优化的。唯一的办法只有将数据缓存起来，定期更新，这适用于结果对实效性要求不高的操作。

事实上，索引是将条件查询、排序的读操作的资源消耗分布到了写操作中，索引越多，耗费磁盘空间越大，写操作越慢。因此，决不能盲目添加索引。对字段索引与否，最根本的依据是 SQL 语句执行的概率、表的大小和写操作的频繁程度。

A.3.8　注释文档

程序注释中要用简单、明了、精确的语言表达出程序的处理说明。应避免每行程序都使用注释，可以在一段程序的前面加一段注释，要具有明确的处理逻辑。注释虽然必不可少，但也不应过多，不要被动地为写注释而写注释。

1. 文件

每个包含 PHP 代码的文件必须至少在文件顶部包含这些 phpDocumentor 标签：

```
/**
 * 文件的简短描述
 *
 * 文件的详细描述（如果有的话）... ...
 *
```

5．单行注释

短注释可以显示在一行内，并与其后的代码具有一样的缩进层级。如果一个注释不能在一行内写完，就应采用块注释。单行注释之前应该有一个空行。以下是一个代码中单行注释的例子：

```
If (condition) {
  /* 以下代码运行的条件 */
  ...
}
```

6．尾端注释

极短的注释可以与其所要描述的代码位于同一行，但是应该有足够的空白来分开代码和注释。若有多个短注释出现于大段代码中，它们应该具有相同的缩进。

以下是一个代码中尾端注释的例子：

```
If ($a == 2) {
      return TRUE; /* 对单一条件的说明 */
} Else {
      return isPrime($a); /* 其余的条件 */
}
```

7．行末注释

注释界定符“//”，可用于注释整行或者一行中的一部分。

```
If ($Euspice > 1) {
    ...      //注释用法
}
```